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 № 3

на тему :

„ ПОЛІМОРФІЗМ 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](#_Toc497851311)

[2 ПОСТАНОВКА ЗАДАЧІ 4](#_Toc497851312)

[3 UML ДІАГРАМА 5](#_Toc497851313)

[4 ПРОГРАМНИЙ КОД 6](#_Toc497851314)

[Main.cpp 6](#_Toc497851315)

[Figure.cpp 6](#_Toc497851316)

[Figure.hpp 6](#_Toc497851317)

[Cylinder.cpp 7](#_Toc497851318)

[Cylinder.hpp 7](#_Toc497851319)

[Cone.cpp 7](#_Toc497851320)

[Cone.hpp 8](#_Toc497851321)

[CutCone.cpp 8](#_Toc497851322)

[CutCone.hpp 8](#_Toc497851323)

[5 РЕЗУЛЬТАТ РОБОТИ 9](#_Toc497851324)

[6 ВИСНОВОК 10](#_Toc497851325)

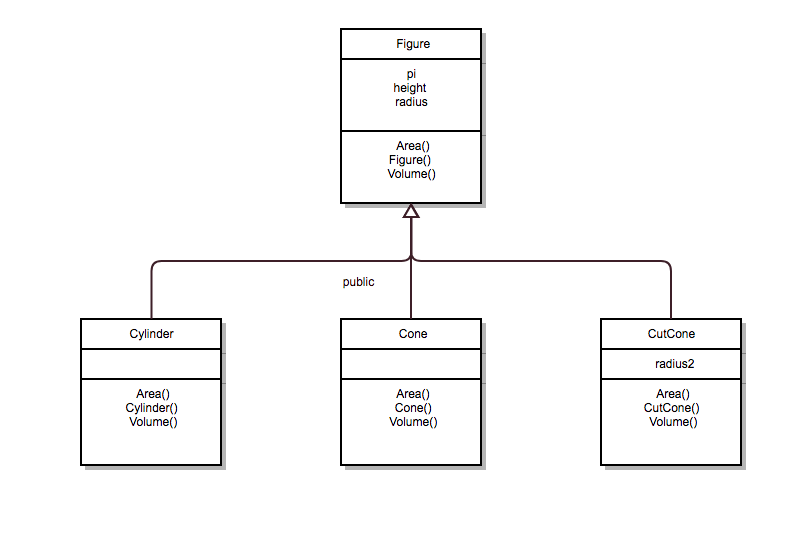
# ЦІЛЬ РОБОТИ

Мета роботи - вивчити основні концепції об'єктно-орієнтованого програмування.Вивчити особливості віртуальних функцій,абстрактних класів і поліморфізму.

# ПОСТАНОВКА ЗАДАЧІ

Спроектувати ієрархію класів: клас фігури в просторі і його спадкоємці: циліндр, конус, усічений конус. Визначити в базовому класі і перевизначити в спадкоємців методи обчислення повної площі поверхні і об'єму фігури. Елементи-дані фігур оголошуються в базовому класі, а не започатковано в спадкоємців (елементи дані: радіуси і висота).

# UML ДІАГРАМА



# ПРОГРАМНИЙ КОД

## Main.cpp

# #include <iostream>

#include "Cone.hpp"

#include "CutCone.hpp"

#include "Сylinder.hpp"

using namespace std;

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

    cout<<"Цицилюк Анна ІП-61 Варіант 19 Рівень В"<<endl<<"Спроектировать иерархию классов: класс фигуры в пространстве и его наследники: цилиндр, конус, усеченный конус. Определить в базовом классе и переопределить в наследниках методы вычисления полной площади поверхности и объема фигуры. Элементы-данные фигур объявляются в базовом классе, а инициализируются в наследниках (элементы данные: радиусы и высота)."<<endl;

    int action;

    do{

    cout<<"Що створити?(1-циліндр, 2-конус ,3 -зрізаний конус)або введіть 0 для виходу"<<endl;

    cin>>action;

    if(action==1)

    {  Cylinder F;

        cout<<"Площа "<< F.Area()<<endl;

        cout<<"Об'єм "<<F.Volume()<<endl;

    }

        if(action==2)

        {  Cone F;

            cout<<"Площа "<< F.Area()<<endl;

            cout<<"Об'єм "<<F.Volume()<<endl;

        }

        if(action==3)

        {  CutCone F;

            cout<<"Площа "<< F.Area()<<endl;

            cout<<"Об'єм "<<F.Volume()<<endl;

        }

    }

    while (action!=0);

    return 0;

}

## Figure.cpp

#include "Figure.hpp"

Figure::Figure()

{pi=3.1415926;};

## Figure.hpp

#ifndef Figure\_hpp

#define Figure\_hpp

#include <iostream>

#include <stdio.h>

class Figure

{

protected:

    float pi;

    float radius;

    float height;

public:

    Figure();

    virtual float Area()=0;

    virtual float Volume()=0;

};

#endif /\* Figure\_hpp \*/

## Cylinder.cpp

#include "Сylinder.hpp"

Cylinder::Cylinder()

{

    do{

    std::cout<<"Введіть радіус та висоту циліндра"<<std::endl;

    std::cin>>radius>>height;

    }

    while(radius<0&&height<0);

}

float Cylinder:: Area()

{

    float S;

    S=(2\*pi\*radius\*height)+(2\*pi\*radius\*radius);

    return S;

}

float Cylinder::Volume()

{

    float V;

    V=pi\*radius\*radius\*height;

    return V;

}

## Cylinder.hpp

#ifndef \_ylinder\_hpp

#define \_ylinder\_hpp

#include "Figure.hpp"

#include <stdio.h>

class Cylinder:public Figure

{

public:

    Cylinder();

    float Area();

    float Volume();

};

#endif /\* \_ylinder\_hpp \*/

## Cone.cpp

#include "Cone.hpp"

Cone::Cone()

{

    do{

    std::cout<<"Введіть радіус та висоту конуса"<<std::endl;

    std::cin>>radius>>height;

    }

     while(radius<0&&height<0);

}

float Cone:: Area()

{

    float S;

    float l=2\*pi\*radius;

    S=pi\*radius\*(radius+l);

    return S;

}

float Cone::Volume()

{

    float V;

    V=(pi\*radius\*radius\*height)/3;

    return V;

}

## Cone.hpp

#ifndef Cone\_hpp

#define Cone\_hpp

#include "Figure.hpp"

#include <stdio.h>

class Cone:public Figure

{

public:

    Cone();

    float Area();

    float Volume();

};

#endif /\* Cone\_hpp \*/

## CutCone.cpp

#include "CutCone.hpp"

CutCone::CutCone()

{

    do{

    std::cout<<"Введіть радіус 1 та 2 та висоту зрізаного конуса"<<std::endl;

    std::cin>>radius>>radius2>>height;

    }

     while(radius<0&&height<0&&radius2<0);

}

float CutCone:: Area()

{

    float S;

    float l=2\*pi\*radius;

    S=pi\*(radius\*radius+(radius2+radius)\*l+radius2\*radius2);

    return S;

}

float CutCone::Volume()

{

    float V;

    V=(pi\*height/3)\*(radius\*radius+radius2\*radius+radius2\*radius2);

    return V;

}

## CutCone.hpp

#ifndef CutCone\_hpp

#define CutCone\_hpp

#include "Figure.hpp"

#include <stdio.h>

class CutCone:public Figure

{

private:float radius2;

public:

    CutCone();

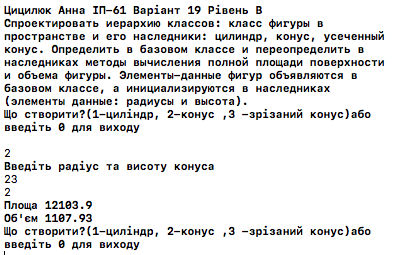
    float Area();

    float Volume();

};

#endif /\* CutCone\_hpp \*/

# РЕЗУЛЬТАТ РОБОТИ



# ВИСНОВОК

Я вивчила основні концепції об'єктно-орієнтованого програмування.Також засвоїла знання про особливості віртуальних функцій,абстрактних класів і поліморфізму.