Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №5 з дисципліни
«Основи програмування 2.
Модульне програмування»

«Успадкування та поліморфізм»

Варіант 32

Виконав студент ІП-13 Шевцова Анастасія Андріївна
(шифр, прізвище, ім'я, по батькові)

Перевірив Вєчерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №5

Успадкування та поліморфізм

Варіант 32

Спроектувати клас TFigure, який представляє просторову геометричну фігуру з методами обчислення площі її поверхні та об’єму. На основі цього класу створити класи-нащадки TPyramid та TCylinder. Створити n пірамід і m циліндрів. Знайти циліндр з найбільшим об’ємом і піраміду – з найменшою площею поверхні.

C#

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab_5_cs
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Кількість пірамід: ");
            int n = int.Parse(Console.ReadLine());
            n = (int)Methods.CheckValue(n);
            TPyramid[] pyramids = new TPyramid[n];
            Methods.MakePyramidsArray(pyramids);

            Console.WriteLine("-----------------------");

            Console.Write("Кількість циліндрів: ");
            int m = int.Parse(Console.ReadLine());
            m = (int)Methods.CheckValue(m);
            TCylinder[] cylinders = new TCylinder[m];
            Methods.MakeCylinderArray(cylinders);

            int indC = Methods.BiggestVolume(cylinders);
            int indP = Methods.LeastSurfaceArea(pyramids);

            Console.WriteLine("\nЦиліндр з найбільшим об'ємом: {0}, об'єм = {1,
0:0.###}", indC + 1, cylinders[indC].FindVolume());
            Console.WriteLine("Піраміда з найменшою площою поверхні: {0}, площа пове-
рхні = {1, 0:0.###}", indP + 1, pyramids[indP].FindSurfaceArea());
        }
    }
}
```

TFigure.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace lab_5_cs
{
    abstract class TFigure
    {
        protected double _height;
        public double Height { set { _height = value; } }
        public abstract double FindSurfaceArea();
        public abstract double FindVolume();
    }
}
```

TPyramid.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab_5_cs
{
    class TPyramid : TFigure
    {
        private double _apopheme;
        private double _baseSide;
        public double Apopheme {set { _apopheme = value; } }
        public double BaseSide {set { _baseSide = value; } }

        public override double FindSurfaceArea()
        {
            double sideSurfaceArea = 0.5 * (4 * _baseSide) * _apopheme;
            double baseArea = Math.Pow(_baseSide, 2);
            return sideSurfaceArea + baseArea;
        }

        public override double FindVolume()
        {
            return (1 / 3.0) * Math.Pow(_baseSide, 2) * _height;
        }

        public bool CheckData()
        {
            if (Math.Pow(_apopheme, 2) == Math.Pow(_height, 2) +
Math.Pow(_baseSide/2.0, 2))
            {
                return true;
            }
            else { return false; }
        }
    }
}
```

TCylinder.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab_5_cs
{
    class TCylinder : TFigure
    {
        private double _radius;
        public double Radius {set { _radius = value; } }
```

```
        public override double FindSurfaceArea()
        {
            double sideSurfaceArea = 2 * Math.PI * _radius * _height;
            double baseArea = Math.PI * Math.Pow(_radius, 2);
            return sideSurfaceArea + baseArea;
        }

        public override double FindVolume()
        {
            return Math.PI * Math.Pow(_radius, 2) * _height;
        }
    }
}
```

Methods.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab_5_cs
{
    internal static class Methods
    {
        public static int BiggestVolume(TCylinder[] cylinders)
        {
            int index = 0;
            double max = cylinders[index].FindVolume();
            for (int i = 0; i < cylinders.Length; i++)
            {
                if (cylinders[i].FindVolume() > max)
                {
                    max = cylinders[i].FindVolume();
                    index = i;
                }
            }
            return index;
        }

        public static int LeastSurfaceArea(TPyramid[] pyramids)
        {
            int index = 0;
            double min = pyramids[index].FindVolume();
            for (int i = 0; i < pyramids.Length; i++)
            {
                if (pyramids[i].FindVolume() < min)
                {
                    min = pyramids[i].FindVolume();
                    index = i;
                }
            }
            return index;
        }

        public static void MakePyramidsArray(TPyramid[] pyramids)
        {
            for (int i = 0; i < pyramids.Length; i++)
            {
                do
                {
                    Console.WriteLine($"\n{i + 1} піраміда:");

                    pyramids[i] = new TPyramid();
```

```
                    Console.Write("Висота піраміди: ");
                    double height = double.Parse(Console.ReadLine());
                    height = CheckValue(height);
                    pyramids[i].Height = height;

                    Console.Write("Апофема: ");
                    double apopheme = double.Parse(Console.ReadLine());
                    apopheme = CheckValue(apopheme);
                    pyramids[i].Apopheme = apopheme;

                    Console.Write("Сторорна основи: ");
                    double baseSide = double.Parse(Console.ReadLine());
                    baseSide = CheckValue(baseSide);
                    pyramids[i].BaseSide = baseSide;

                    if (!pyramids[i].CheckData())
                    {
                        Console.WriteLine("-----Некоректні дані! Спробуйте знову.----
-");
                    }
                }
                while (!pyramids[i].CheckData());
            }
        }

        public static void MakeCylinderArray(TCylinder[] cylinders)
        {
            for (int i = 0; i < cylinders.Length; i++)
            {
                Console.WriteLine($"\n{i + 1} циліндр:");

                cylinders[i] = new TCylinder();

                Console.Write("Радіус: ");
                double radius = double.Parse(Console.ReadLine());
                radius = CheckValue(radius);
                cylinders[i].Radius = radius;

                Console.Write("Висота циліндра: ");
                double height = double.Parse(Console.ReadLine());
                height = CheckValue(height);
                cylinders[i].Height = height;
            }
        }

        public static double CheckValue(double value)
        {
            while(value < 0)
            {
                Console.WriteLine("Неправильно! Число має бути > 0");
                Console.Write("Спробуйте ще раз: ");
                value = double.Parse(Console.ReadLine());
            }
            return value;
        }
    }
}
```

Тестування

```
Кількість пірамід: 2

1 піраміда:
Висота піраміди: 3
Апофема: 5
Сторорна основи: 8

2 піраміда:
Висота піраміди: 24
Апофема: 26
Сторорна основи: 20
------------------------
Кількість циліндрів: 3

1 циліндр:
Радіус: 4,8
Висота циліндра: 10

2 циліндр:
Радіус: 5
Висота циліндра: 20

3 циліндр:
Радіус: 15,7
Висота циліндра: 15

Циліндр з найбільшим об'ємом: 3, об'єм = 11615,568
Піраміда з найменшою площою поверхні: 1, площа поверхні = 144
Press any key to continue . . .
```

Python

lab_5_py.py

```
from methods import *

n = int(input("Кількість пірамід: "))
n = int(check_value(n))
pyramids = []
pyramids = make_pyramids_array(pyramids, n)

print("------------------------")

m = int(input("Кількість циліндрів: "))
m = int(check_value(m))
cylinders = []
cylinders = make_cylinder_array(cylinders, m)

ind_c = biggest_volume(cylinders)
ind_p = least_surface_area(pyramids)

print("\nЦиліндр з найбільшим об'ємом: {0}, об'єм = {1:10.3f}".format(ind_c+1,
cylinders[ind_c].find_volume()))
print("Піраміда з найменшою площою поверхні: {0}, площа поверхні =
{1:10.3f}".format(ind_p + 1, pyramids[ind_p].find_surface_area()))
```

TFigure.py

```
from abc import ABC, abstractmethod

class TFigure(ABC):
    _Pi = 3.1416
```

```
    _height = 0.0

    @abstractmethod
    def find_surface_area(self):
        pass
    def find_volume(self):
        pass
```

TPyramid.py

```
from TFigure import *

class TPyramid(TFigure):

    __apopheme = 0.0
    __base_side = 0.0

    def __init__(self, pyramid):
        self._height = float(pyramid[0])
        self.__apopheme = float(pyramid[1])
        self.__base_side = float(pyramid[2])

    def find_surface_area(self):
        side_surface_area = 0.5 * (4 * self.__base_side) * self.__apopheme
        base_area = pow(self.__base_side, 2)
        return side_surface_area + base_area

    def find_volume(self):
        return (1/3.0) * pow(self.__base_side, 2) * self._height

    def check_data(self):
        if pow(self.__apopheme, 2) == pow(self._height, 2) +
pow(self.__base_side/2.0, 2):
            return True
        else:
            return False
```

TCylinder.py

```
from TFigure import *

class TCylinder(TFigure):

    __radius = 0.0

    def __init__(self, cylinder):
        self.__radius = float(cylinder[0])
        self._height = float(cylinder[1])

    def find_surface_area(self):
        side_surface_area = 2 * self._Pi * self.__radius * self._height
        base_area = self._Pi * pow(self.__radius, 2)
        return side_surface_area + base_area

    def find_volume(self):
        return self._Pi * pow(self.__radius, 2) * self._height
```

methods.py

```
from TPyramid import *
from TCylinder import *

def biggest_volume(cylinders):
```

```
    index = 0
    maximum = cylinders[index].find_volume()
    for i in range(len(cylinders)):
        if (cylinders[i].find_volume() > maximum):
            maximum = cylinders[i].find_volume()
            index = i
    return index

def least_surface_area(pyramids):
    index = 0
    minimum = pyramids[index].find_volume()
    for i in range(len(pyramids)):
        if (pyramids[i].find_volume() < minimum):
            minimum = pyramids[i].find_volume()
            index = i
    return index

def check_value(value):
    while value < 0:
        print("Неправильно! Число має бути > 0")
        value = float(input("Спробуйте ще раз: "))
    return value

def make_pyramids_array(pyramids, size):
    for i in range(size):
        while True:
            print(f"\n{i + 1} піраміда:")
            pyramid = []

            height = float(input("Висота піраміди: "))
            height = check_value(height)
            pyramid.append(height)

            apopheme = float(input("Апофема: "))
            apopheme = check_value(apopheme)
            pyramid.append(apopheme)

            base_side = float(input("Сторорна основи: "))
            base_side = check_value(base_side)
            pyramid.append(base_side)

            pyramids.append(TPyramid(pyramid))
            if pyramids[i].check_data() == False:
                print("-----Некоректні дані! Спробуйте знову.-----")
            if pyramids[i].check_data() == True:
                break

    return pyramids

def make_cylinder_array(cylinders, size):
    for i in range(size):
        print(f"\n{i + 1} циліндр:")
        cylinder = []

        radius = float(input("Радіус: "))
        radius = check_value(radius)
        cylinder.append(radius)

        height = float(input("Висота циліндра: "))
        height = check_value(height)
        cylinder.append(height)

        cylinders.append(TCylinder(cylinder))
    return cylinders
```

Тестування

```
Кількість пірамід: 2

1 піраміда:
Висота піраміди: 3
Апофема: 5
Сторорна основи: 8

2 піраміда:
Висота піраміди: 24
Апофема: 26
Сторорна основи: 20
------------------------
Кількість циліндрів: 3

1 циліндр:
Радіус: 4.8
Висота циліндра: 10

2 циліндр:
Радіус: 5
Висота циліндра: 20

3 циліндр:
Радіус: 15.7
Висота циліндра: 15

Циліндр з найбільшим об'ємом: 3, об'єм =  11615.595
Піраміда з найменшою площою поверхні: 1, площа поверхні =    144.000
Press any key to continue . . .
```