

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.  
Методології програмування» «Поліморфізм та наслідування»

Варіант 31

**Виконав студент** ІІ-13 Сокур Антон Юрійович  
(шифр, прізвище, ім'я, по батькові)

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

Київ 2022

**Мета** – вивчити механізми створення класів з використанням поліморфізму та наслідування.

### Завдання:

31. Створити клас TBody, який представляє просторову геометричну фігуру з методами обчислення площі її поверхні та об'єму. На основі цього класу створити класи-нащадки TParallelepiped та TBall. Створити певну кількість паралелепіпедів та куль, щоб їх сумарна кількість дорівнювала n. Для куль знайти сумарний об'єм, а для паралелепіпедів – суму площ поверхні.

### Код програми

## C#

### TBody.cs

```
using System;

namespace ConsoleApp1
{
    class TBody
    {
        private double Side1, Side2, Side3;

        public double side1
        {
            get { return Side1; }
            set { Side1 = value; }
        }
        public double side2
        {
            get { return Side2; }
            set { Side2 = value; }
        }
        public double side3
        {
            get { return Side3; }
            set { Side3 = value; }
        }

        public virtual double CountS()
        {
            Console.WriteLine("Program is not able to count area of this figure!");
            return -1;
        }
        public virtual double CountV()
        {
            Console.WriteLine("Program is not able to count volume of this figure!");
            return -1;
        }
    }
}
```

### TBall.cs

```
using System;

namespace ConsoleApp1
```

```

{
    class TBall : TBody
    {
        public TBall(bool Key = false)
        {
            side3 = -1;
            side2 = -1;
            if (Key)
            {
                Random rand = new Random();
                side1 = rand.NextDouble() * 15 + 0.01;
            }
        }
        public override double CountS()
        {
            double S;
            S = 4 * Math.PI * Math.Pow(side1, 2);
            return S;
        }
        public override double CountV()
        {
            double V;
            V = 4 * Math.PI * Math.Pow(side1, 3) / 3;
            return V;
        }
    }
}

```

## TParallelepiped.cs

```

using System;

namespace ConsoleApp1
{
    class TParallelepiped : TBody
    {
        public TParallelepiped(bool Key = false)
        {
            if (Key)
            {
                Random rand = new Random();
                var maxValue = 15; // Max rand value
                side1 = rand.NextDouble() * maxValue + 0.01;
                side2 = rand.NextDouble() * maxValue + 0.01;
                side3 = rand.NextDouble() * maxValue + 0.01;
            }
        }

        public override double CountS()
        {
            double S = 0;
            S = 2 * (side1 * side2 + side2 * side3 + side3 * side1);
            return S;
        }

        public override double CountV()
        {
            double V;
            V = side1 * side2 * side3;
            return V;
        }
    }
}

```

```
}  
}
```

## Program.cs

```
using System;  
  
namespace ConsoleApp1  
{  
  
    internal class Program  
    {  
        static double SumV(TBall[] balls, int n)  
        {  
            double result = 0;  
            for(int i = 0; i < n; i++)  
            {  
                result += balls[i].CountV();  
            }  
            return result;  
        }  
  
        static double SumS(TParallelepiped[] parallelepipeds, int n)  
        {  
            double result = 0;  
            for (int i = 0; i < n; i++)  
            {  
                result += parallelepipeds[i].CountS();  
            }  
            return result;  
        }  
        static void InfoBalls(TBall[] balls, int n)  
        {  
            for (int i = 0; i < n; i++)  
            {  
                Console.WriteLine($"Ball number {i + 1}");  
                Console.WriteLine($"R = {balls[i].side1} , S = {balls[i].CountS()}, V  
= {balls[i].CountV()}");  
                Console.WriteLine();  
            }  
        }  
  
        static void InfoParallelepiped(TParallelepiped[] parallelepipeds, int n)  
        {  
            for (int i = 0; i < n; i++)  
            {  
                Console.WriteLine($"Parallelepiped number {i + 1}");  
                Console.WriteLine($"a = {parallelepiped[i].side1} , b =  
{parallelepiped[i].side2} , c = {parallelepiped[i].side3} , S =  
{parallelepiped[i].CountS()}, V = {parallelepiped[i].CountV()}");  
                Console.WriteLine();  
            }  
        }  
  
        static TBall[] setBalls(int n)  
        {  
            TBall[] balls = new TBall[n];  
            for(int i = 0; i < n; i++)  
            {  
                Console.WriteLine("Do you want to generate ball automatically  
(yes/no)?");
```

```

        if(Console.ReadLine() == "yes")
        {
            balls[i] = new TBall(true);
        }
        else
        {
            balls[i] = new TBall();
            Console.Write("Input radius:");
            double R = double.Parse(Console.ReadLine());
            balls[i].side1 = R;
        }
    }
    return balls;
}
static TParallelepiped[] setParallelepipedes(int m)
{
    TParallelepiped[] parallelepipedes = new TParallelepiped[m];
    for (int i = 0; i < m; i++)
    {
        Console.WriteLine("Do you want to generate parallelepiped
automatically (yes/no)?");
        if (Console.ReadLine() == "yes")
        {
            parallelepipedes[i] = new TParallelepiped(true);
        }
        else
        {
            parallelepipedes[i] = new TParallelepiped();
            Console.Write("Input side1:");
            double a = double.Parse(Console.ReadLine());
            parallelepipedes[i].side1 = a;
            Console.Write("Input side2:");
            double b = double.Parse(Console.ReadLine());
            parallelepipedes[i].side2 = b;
            Console.Write("Input side3:");
            double c = double.Parse(Console.ReadLine());
            parallelepipedes[i].side3 = c;
        }
    }
    return parallelepipedes;
}
static void Main(string[] args)
{
    int n;
    Console.WriteLine("Input number of figures:");
    n = int.Parse(Console.ReadLine());
    Random rand = new Random();
    int m = rand.Next(1, n-1);
    n -= m;
    Console.WriteLine($"We will create {n} ball and {m} parallelepipedes!");
    TBall[] balls = setBalls(n);
    TParallelepiped[] parallelepipedes = setParallelepipedes(m);
    Console.WriteLine();
    InfoBalls(balls, n);
    InfoParallelepipedes(parallelepipedes, m);
    Console.WriteLine($"Sum of balls' volumes = {SumV(balls, n)}");
    Console.WriteLine($"Sum of parallelepipedes' areas =
{SumS(parallelepipedes, m)}");
}
}
}

```

# Робота програми на C#

```
Input number of figures:4
We will create 2 ball and 2 parallelepipeds!
Do you want to generate ball automatically (yes/no)?yes
Do you want to generate ball automatically (yes/no)?no
Input radius:5,3
Do you want to generate parallelepiped automatically (yes/no)?yes
Do you want to generate parallelepiped automatically (yes/no)?no
Input side1:2,1
Input side2:3
Input side3:5,124

Ball number 1
R = 11,281302581529769 , S = 1599,294190473825, V = 6014,040559872642

Ball number 2
R = 5,3 , S = 352,98935055734916, V = 623,6145193179834

Parallelepiped number 1
a = 0,41814436793444376 , b = 7,060659879494171 , c = 10,776857655039853 , S = 167,1007679339136, V = 31,817326870658096

Parallelepiped number 2
a = 2,1 , b = 3 , c = 5,124 , S = 64,8648, V = 32,2812

Sum of balls' volumes = 6637,655079190626
Sum of parallelepipeds' areas = 231,9655679339136
```

## Python

### classes.py

```
import math
import random

class TBody:
    def __init__(self):
        self.side1 = None
        self.side2 = None
        self.side3 = None

    def countS(self):
        print("Program is not able to count area of this figure!")
        return -1

    def countV(self):
        print("Program is not able to count volume of this figure!")
        return -1

class TBall(TBody):
    def __init__(self, key=False):
        super().__init__()

        self.side3 = -1
        self.side2 = -1

        if key:
            self.side1 = random.random() * 10

    def countS(self):
```

```

        s = 4 * math.pi * self.side1**2
        return s

    def countV(self):
        v = 4 * math.pi * self.side1**3 / 3
        return v

class TParallelepiped(TBody):
    def __init__(self, key=False):
        super().__init__()

        if key:
            self.side1 = random.random() * 15
            self.side2 = random.random() * 15
            self.side3 = random.random() * 15

    def countS(self):
        s = 2 * (self.side1 * self.side2 + self.side2 * self.side3 +
self.side3 * self.side1)
        return s

    def countV(self):
        v = self.side1 * self.side2 * self.side3
        return v

```

## functions.py

```

from classes import TBall, TParallelepiped

def sumV(balls, n):
    result = 0
    for i in range(n):
        result += balls[i].countV()
    return result

def sumS(parallelepipeds, n):
    result = 0
    for i in range(n):
        result += parallelepipeds[i].countS()
    return result

def info_balls(balls, n):
    for i in range(n):
        print(f"Ball number {i + 1}")
        print(f"R = {balls[i].side1} , S = {balls[i].countS()} , V =
{balls[i].countV()}")
        print()

def info_parallelepipeds(parallelepipeds, n):
    for i in range(n):
        print(f"Parallelepiped number {i + 1}")
        print(
            f"a = {parallelepipeds[i].side1} , b = {parallelepipeds[i].side2}
, c = {parallelepipeds[i].side3} , "
            f"S = {parallelepipeds[i].countS()} , V =
{parallelepipeds[i].countV()}")
        print()

```

```

def set_balls(n):
    balls = []
    for _ in range(n):
        ans = input("Do you want to generate ball automatically (yes/no)?")
        if ans == "yes":
            balls.append(TBall(True))
        else:
            ball = TBall()
            ball.side1 = float(input("Input radius: "))
            balls.append(ball)
    return balls

def set_parallelepipedes(n):
    parallelepipedes = []
    for _ in range(n):
        ans = input("Do you want to generate parallelepiped automatically (yes/no)?")
        if ans == "yes":
            parallelepipedes.append(TParallelepiped(True))
        else:
            parallelepiped = TParallelepiped()
            parallelepiped.side1 = float(input("Input side1: "))
            parallelepiped.side2 = float(input("Input side2: "))
            parallelepiped.side3 = float(input("Input side3: "))
            parallelepipedes.append(parallelepiped)
    return parallelepipedes

```

## main.py

```

import random
from functions import set_balls, set_parallelepipedes, info_balls,
info_parallelepipedes, sumV, sumS

def main():
    n = int(input("Input number of figures: "))
    m = random.randint(1, n - 1)
    n -= m
    print(f"We'll create {n} ball and {m} parallelepipedes!")
    balls = set_balls(n)
    parallelepipedes = set_parallelepipedes(m)
    print()
    info_balls(balls, n)
    info_parallelepipedes(parallelepipedes, m)
    print(f"Sum of balls' volumes = {sumV(balls, n)}")
    print(f"Sum of parallelepipedes' areas = {sumS(parallelepipedes, m)}")

if __name__ == '__main__':
    main()

```



# Робота програми на Python

```
C:\Users\anton\Python\Python310\python.exe D:/study/OP_2_sem/lab4/python/main.py
Input number of figures: 4
We'll create 2 ball and 2 parallelepipeds!
Do you want to generate ball automatically (yes/no)?yes
Do you want to generate ball automatically (yes/no)?no
Input radius: 3.12
Do you want to generate parallelepiped automatically (yes/no)?yes
Do you want to generate parallelepiped automatically (yes/no)?no
Input side1: 4.5
Input side2: 2
Input side3: 8.3

Ball number 1
R = 0.9751179082684513 , S = 11.948795514027115, V = 3.883828162655192

Ball number 2
R = 3.12 , S = 122.32607810841795, V = 127.21912123275466

Parallelepiped number 1
a = 8.276645800882566 , b = 5.0613583953640155 , c = 12.508281990628314 , S = 417.4531767771746, V = 523.9853253269195

Parallelepiped number 2
a = 4.5 , b = 2.0 , c = 8.3 , S = 125.9, V = 74.7

Sum of balls' volumes = 131.10294939540984
Sum of parallelepipeds' areas = 543.3531767771746
```

**Висновок:** під час виконання даної лабораторної роботи я вивчив особливості створення класів з використання поліморфізму та наслідування у мові C#