ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ

КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

**Отчет по лабораторной работе 6**

**по дисциплине: «Новые информационные технологии»**

студента очного отделения

4 курса 12001801 группы

Капустина Виктора Сергеевича

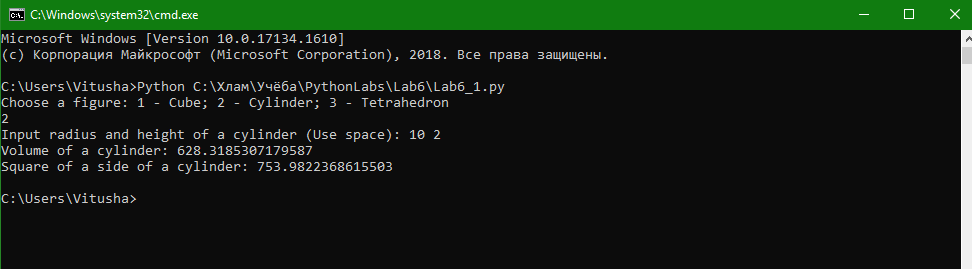
Проверил(а):

Батищев Денис Сергеевич

Белгород 2021

**Задание**1. На основании предложенной предметной области спроектировать 3-4 класса, используя механизм наследования. Для каждого класса использовать отдельный модуль.  
 2. Предусмотреть у класса наличие полей, методов и свойств. Названия членов класса должны быть осмысленны и снабжены комментариями.   
3. Один из наследников должен перегружать метод родителя.  
 4. Один из классов должен содержать виртуальный метод, который переопределяется в одном наследнике и не переопределяется в другом.   
5. Продемонстрировать работу всех объявленных методов.   
6. Продемонстрировать вызов конструктора родительского класса при наследовании.

**Индивидуальное задание:** Написать программу, в которой описана иерархия классов: геометрические фигуры (куб, цилиндр, тетраэдр). Реализовать методы вычисления объема и площади поверхности Реализовать методы вычисления объема и площади поверхности фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

  
Рис. 1. Результат работы программы

**Листинг программы 1**

import math

import Attributes

import Cube

import Cylinder

import Tetrahedron

class Attributes:

V = 0

S = 0

def \_\_init\_\_(self, a):

self.a = a

def GetV(self):

return(self.V)

def GetS(self):

return(self.S)

class Cube(Attributes.Attributes):

def Volume(self):

self.V = self.a\*\*3

def Square(self):

self.S = 6 \* self.a\*\*2

class Cylinder(Cube.Cube):

#конструктор класса цилиндр

def \_\_init\_\_(self, r, h):

self.r, self.h = r, h

def Volume(self):

self.V = math.pi \* self.r\*\*2 \* self.h

def Square(self):

self.S = 2 \* math.pi \* self.r \* (self.h + self.r)

class Tetrahedron(Attributes.Attributes):

def Volume(self):

self.V = 2 / (12 \* self.a)

def Square(self):

self.S = math.sqrt(3 \* self.a\*\*2)

change = int(input('Choose a figure: 1 - Cube; 2 - Cylinder; 3 - Tetrahedron\n'))

if(change == 1):

a = float(input('Input length of a cube side: '))

cube = Cube.Cube(a)

cube.Volume()

cube.Square()

print('Volume of a cube:', cube.GetV(), '\nSquare of a side of a cube:', cube.GetS())

elif(change == 2):

r,h = map(float,input('Input radius and height of a cylinder (Use space):').split())

cylinder = Cylinder.Cylinder(r,h)

cylinder.Volume()

cylinder.Square()

print('Volume of a cylinder:', cylinder.GetV(), '\nSquare of a side of a cylinder:', cylinder.GetS())

elif(change == 3):

a = float(input('Input a length of a side of Tetrahedron: '))

tetrahedron = Tetrahedron.Tetrahedron(a)

tetrahedron.Volume()

tetrahedron.Square()

print('Volume of Tetrahedron:', tetrahedron.GetV(), '\nSquare of a Tetrahedron:', tetrahedron.GetS())

**Конец листинга 1**