

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт информационных технологий** | **Кафедра информационных систем** |

**Основная образовательная программа 09.03.02  
«Информационные системы»**

**Отчет по дисциплине «Веб-программирование»**

**по лабораторной работе №3**

**«Основы языка Python»**

|  |  |
| --- | --- |
| Студент  группы ИДБ-19-08 | Деминова Ю.И. |
|  |  |
| Преподаватель  к.т.н., доц. | Адамова Ю.С. |

Содержание

[Задание №1 3](#_Toc88395506)

[Код программы 4](#_Toc88395507)

[Пример работы программы 5](#_Toc88395508)

[Задание №2 5](#_Toc88395509)

[Код программы 6](#_Toc88395510)

[Пример работы программы 8](#_Toc88395511)

GitHub: https://github.com/dielellis/Veb-programmirovanie

Задание №1

Написать класс Sphere для представления сферы в трехмерном

пространстве.

* конструктор, принимающий 4 действительных числа: радиус, и 3 координаты центра шара. Если конструктор вызывается без аргументов, создать объект сферы с единичным радиусом и центром в начале координат.
* метод get\_volume (), который возвращает действительное число — объем шара, ограниченной текущей сферой.
* метод get\_square\_(), который возвращает действительное число - площадь внешней поверхности сферы.
* метод get\_radius\_(), который возвращает действительное число — радиус сферы.
* метод get\_center\_(), который возвращает тьюпл с 3 действительными числами — координатами центра сферы в том же порядке, в каком они задаются в конструкторе.
* метод set\_radius\_(r), который принимает 1 аргумент — действительное число, и меняет радиус текущей сферы, ничего не возвращая.
* метод set\_center (x, y, z), который принимает 3 аргумента — действительных числа, и меняет координаты центра сферы, ничего не возвращая. Координаты задаются в том же порядке, что и в конструкторе.
* метод is\_point\_inside (x, y, z), который принимает 3 аргумента — действительных числа — координаты некоторой точки в пространстве (в том же порядке, что и в конструкторе), и возвращает логическое значение True или False в зависимости от того, находится эта точка внутри сферы.

Код программы

import math

class Sphere:

def \_\_init\_\_(self, r=1, x=0, y=0, z=0):

self.radius = r

self.x = x

self.y = y

self.z = z

def get\_volume(self):

vol = 4/3 \* math.pi \* (self.radius \*\* 3)

return vol

def get\_square\_(self):

square = 4 \* math.pi \* (self.radius \*\* 2)

return square

def get\_radius\_(self):

return self.radius

def get\_center\_(self):

return (x, y, z)

def set\_radius(self, r):

self.radius = r

def set\_center(self, x, y, z):

self.x = x

self.y = y

self.z = z

def is\_point\_inside(self, x, y, z):

dist = (self.x - x) \*\* 2 + (self.y - y) \*\* 2 + (self.z - z) \*\* 2

if self.radius \*\* 2 >= dist:

return True

else:

return False

x = float(input("Введите х: "))

y = float(input("Введите y: "))

z = float(input("Введите z: "))

r = float(input("Введите r: "))

sphere = Sphere(r, x, y, z)

print("Объем шара:", sphere.get\_volume())

print("Площадь внешней поверхности шара:", sphere.get\_square\_())

print("Радиус шара:", sphere.get\_radius\_())

print("Центр шара:", sphere.get\_center\_())

x = float(input("Введите новый х: "))

y = float(input("Введите новый y: "))

z = float(input("Введите новый z: "))

r = float(input("Введите новый r: "))

sphere.set\_center(x, y, z)

sphere.set\_radius(r)

x = float(input("Введите координату х точки: "))

y = float(input("Введите координату y точки: "))

z = float(input("Введите координату z точки: "))

print("Находится ли точка внутри шара", sphere.is\_point\_inside(x, y, z))

Пример работы программы

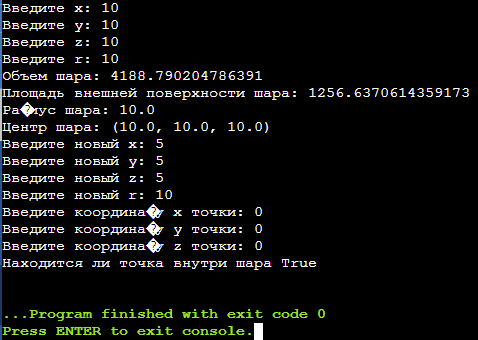


Рис. 1. Пример работы программы

Задание №2

Написать класс Matrix, который будет прообразом математического объекта (для простоты квадратная матрица). Определить для него методы сравнения на основе детерминанта этой матрицы - если детерминант первой матрицы больше, то и матрица больше. Для остальных операций аналогично. Так же реализовать операции сложения, умножения двух матриц.

Код программы

import numpy as np

class Matrix:

def \_\_init\_\_(self, size\_=(3, 3)):

self.arr = np.random.randint(1, 10, size=size\_)

print(self.arr)

def \_\_lt\_\_(self, other): # less then - меньше чем

det1 = np.linalg.det(self.arr)

# print("det1:", det1)

det2 = np.linalg.det(other.arr)

# print("det2:", det2)

if det1 < det2:

return True

else:

return False

def \_\_gt\_\_(self, other): # greater then - больше чем

det1 = np.linalg.det(self.arr)

# print("det1:", det1)

det2 = np.linalg.det(other.arr)

# print("det2:", det2)

if det1 > det2:

return True

else:

return False

def \_\_le\_\_(self, other): # less equal - меньше или равно

det1 = np.linalg.det(self.arr)

# print("det1:", det1)

det2 = np.linalg.det(other.arr)

# print("det2:", det2)

if det1 <= det2:

return True

else:

return False

def \_\_ge\_\_(self, other): # great equal - больше или равно

det1 = np.linalg.det(self.arr)

# print("det1:", det1)

det2 = np.linalg.det(other.arr)

# print("det2:", det2)

if det1 >= det2:

return True

else:

return False

def \_\_eq\_\_(self, other): # equal - равно

det1 = np.linalg.det(self.arr)

# print("det1:", det1)

det2 = np.linalg.det(other.arr)

# print("det2:", det2)

if det1 == det2:

return True

else:

return False

def \_\_ne\_\_(self, other): # not equal - не равно

det1 = np.linalg.det(self.arr)

# print("det1:", det1)

det2 = np.linalg.det(other.arr)

# print("det2:", det2)

if det1 != det2:

return True

else:

return False

def sub(self, other): # Сложение матриц

return self.arr + other.arr

def mul(self, other): # Умножение матриц

return self.arr.dot(other.arr)

m1 = Matrix()

m2 = Matrix()

if m1 < m2:

print("m1 < m2")

if m1 <= m2:

print("m1 <= m2")

if m1 > m2:

print("m1 > m2")

if m1 >= m2:

print("m1 >= m2")

if m1 == m2:

print("m1 == m2")

if m1 != m2:

print("m1 != m2")

print(m1.sub(m2))

print(m1.mul(m2))

Пример работы программы

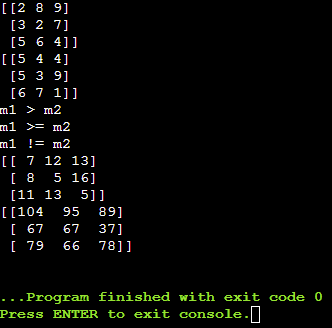


Рис. 2. Пример работы программы