

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

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

**(ФГБОУ ВО МГТУ «СТАНКИН»)**

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

**Отчет по лабораторной работе № 3**

по дисциплине «**Веб-программирование**»

на тему: **Основы языка Python**

|  |  |  |
| --- | --- | --- |
| **Выполнил**  **Студент группы**  ИДБ–19-05 |  | **Клименко Д.А.** |
|  |  |  |
| **Проверил**  преподаватель |  | **Кайшев Д. А.** |
|  |  |  |

Задания:

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:

import math

class Sphere:

def \_\_init\_\_(self, r=1.0, x=0.0, y=0.0, z=0.0):

self.r, self.x, self.y, self.z = r, x, y, z

def get\_volume(self):

volume = 4 / 3.0 \* math.pi \* self.r \*\* 3

return volume

def get\_square(self):

square = 4 \* math.pi \* self.r \*\* 2

return square

def get\_radius(self):

return self.r

def get\_center(self):

return (self.x, self.y, self.z)

def set\_radius(self, r):

self.r = r

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

self.x, self.y, self.z = x, y, z

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

if math.sqrt((self.x - x) \*\* 2 + (self.y - y) \*\* 2 + (self.z - z) \*\* 2) <= self.r:

return True

return False

Код был загружен на GitHub:

https://github.com/DidiKlim/Laborotornaya3