Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего

образования «Московский государственный технический университет имени

Н.Э. Баумана (национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

**Отчет по лабораторной работе № 2 по курсу**

**Базовые компоненты интернет-технологий**

**«Объектно-ориентированные возможности языка Python»**

|  |  |
| --- | --- |
| Выполнил: | Проверил: |
| студент группы ИУ5-34Б |  |
| Харитонов Андрей |  |
| Подпись и дата:  29.12.21 | Подпись и дата: |

Москва, 2021

**Постановка задачи**

1. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.
2. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
3. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
4. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
5. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь.](https://docs.python.org/3/library/abc.html)
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь.](https://docs.python.org/3/library/functions.html#property)
7. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры.
8. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля [math.](https://docs.python.org/3/library/math.html)
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
   * Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
   * Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
10. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов (используйте следующую конструкцию - <https://docs.python.org/3/library/__main__.html>). Создайте следующие объекты и выведите о них информацию в консоль (N - номер Вашего варианта по списку группы):
    * Прямоугольник синего цвета шириной N и высотой N.
    * Круг зеленого цвета радиусом N.
    * Квадрат красного цвета со стороной N.
    * Также вызовите один из методов внешнего пакета, установленного с использованием pip.

Файл “PyLab\_2.py”

from lab\_python\_oop.FigureColor import FigureColor

from lab\_python\_oop.Rectangle import Rectangle

from lab\_python\_oop.Square import Square

from lab\_python\_oop.Circle import Circle

import pygame

def main():

pygame.init()

rectangle = Rectangle(5, 5, FigureColor(0, 0, 255))

square = Square(5, FigureColor(0, 255, 0))

circle = Circle(5, FigureColor(255, 0, 0))

print()

square.Repr()

rectangle.Repr()

circle.Repr()

print()

if \_\_name\_\_ == "\_\_main\_\_":

main()

Пакет lab\_python\_oop

Файл “Figure.py”

from abc import abstractmethod

class Figure():

@abstractmethod

def Square():

pass

Файл “Circle.py”

from lab\_python\_oop.Figure import Figure

from lab\_python\_oop.FigureColor import FigureColor

from math import pi

class Circle(Figure):

name = "Circle"

radius = float()

color = FigureColor(0, 0, 0)

def \_\_init\_\_(self, radius, color):

self.radius = radius

self.color = color

def Square():

return pi \* r\*\*2

def Repr(self):

res = self.name + ":"

res += " radius = {};".format(self.radius)

res += " color = {}.{}.{};".format(self.color.r, self.color.g, self.color.b)

print(res)

Файл “Rectangle.py”

from lab\_python\_oop.Figure import Figure

from lab\_python\_oop.FigureColor import FigureColor

class Rectangle(Figure):

name = "Rectangle"

width = float()

height = float()

color = FigureColor(0, 0, 0)

def \_\_init\_\_(self, width, height, color):

self.color = color

self.width = width

self.height = height

def Square(self):

return self.width \* self.height

def Repr(self):

res = self.name + ":"

res += " width = {};".format(self.width)

res += " height = {};".format(self.height)

res += " color = {}.{}.{};".format(self.color.r, self.color.g, self.color.b)

print(res)

Файл “Square.py”

from lab\_python\_oop.Rectangle import Rectangle

class Square(Rectangle):

name = "Square"

def \_\_init\_\_(self, width, color):

self.width = width

self.height = width

self.color = color

def Repr(self):

res = self.name + ":"

res += " side = {};".format(self.width)

res += " color = {}.{}.{};".format(self.color.r, self.color.g, self.color.b)

print(res)

Файл “FigureColor.py”

class FigureColor():

r = int(0)

g = int(0)

b = int(0)

def \_\_init\_\_(self, r, g, b):

self.r = r

self.g = g

self.b = b

