**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-52Б |  | преподаватель каф. ИУ5 |
| Барышников Михаил |  | Гапанюк Ю.Е. |
|  |  |  |

Москва, 2021 г.

**Описание задания:**

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.

**Текст программы:**

**main.py:**

from lab\_python\_oop.circle import Circle

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.square import Square

import numpy as np

# Press the green button in the gutter to run the script.

if \_\_name\_\_ == '\_\_main\_\_':

rectangle = Rectangle('blue', 3, 3)

circle = Circle('green', 3)

square = Square('red', 3)

rectangle.repr()

circle.repr()

square.repr()

v\_hor\_np = np.array([1, 2])

print(v\_hor\_np)

**figure.py:**

from abc import ABC, abstractmethod

class Figure(ABC):

@abstractmethod

def area(self):

pass

**color.py:**

class Color:

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

self.\_\_color = color

@property

def color(self):

return self.\_\_color

@color.setter

def color(self, color):

self.\_\_color = color

**rectangle.py:**

import figure as f

import color as c

class Rectangle(f.Figure):

figure\_type = "Rectangle"

@classmethod

def get\_figure\_type(cls):

return cls.figure\_type

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

super().\_\_init\_\_()

self.\_\_color = c.Color(color)

self.\_\_width = width

self.\_\_height = height

@property

def color(self):

return self.\_\_color

@color.setter

def color(self, color):

self.\_\_color = c.Color(color)

@property

def width(self):

return self.\_\_width

@width.setter

def width(self, width):

if width > 0:

self.\_\_width = width

else:

raise ValueError

@property

def height(self):

return self.\_\_height

@height.setter

def height(self, height):

if height > 0:

self.\_\_height = height

else:

raise ValueError

def area(self):

return self.\_\_width \* self.\_\_height

def repr(self):

print('FigureType: {}, width: {}, height: {}, color: {}, area: {}!'.format(self.get\_figure\_type, self.\_\_width, self.\_\_height, self.\_\_color.color, self.area()))

**circle.py:**

from figure import Figure

from color import Color

import math

class Circle(Figure):

figure\_type = "Circle"

@classmethod

def get\_figure\_type(cls):

return cls.figure\_type

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

super().\_\_init\_\_()

self.\_\_color = Color(color)

self.\_\_radius = radius

@property

def color(self):

return self.\_\_color

@color.setter

def color(self, color):

self.\_\_color = Color(color)

@property

def radius(self):

return self.\_\_radius

@radius.setter

def radius(self, radius):

if radius > 0:

self.\_\_radius = radius

else:

raise ValueError

def area(self):

return math.pi \* self.\_\_radius \*\* 2

def repr(self):

print('FigureType: {}, radius: {}, color: {}, area: {}!'.format(self.get\_figure\_type, self.\_\_radius, self.\_\_color.color, self.area()))

**Square.py:**

from rectangle import Rectangle

class Square(Rectangle): #Violation SOLID principles!!!

figure\_type = "Square"

@classmethod

def get\_figure\_type(cls):

return cls.figure\_type

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

super().\_\_init\_\_(color, side, side)

def side(self):

return self.\_Rectangle\_\_height

def side(self, side):

if side > 0:

self.\_\_height = side

self.\_\_width = side

else:

raise ValueError

def repr(self):

print('FigureType: {}, side: {}, color: {}, area: {}!'.format(self.get\_figure\_type(), self.\_Rectangle\_\_height, self.color.color, self.area()))

**Экранные формы с примерами выполнения программы:**

