**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

Выполнил: Проверил:

студент группы ИУ5-51Б преподаватель каф. ИУ5

Гапчук Л.Д. Гапанюк Ю.Е.

Подпись и дата: Подпись и дата:

г. Москва, 2020 г.

**Задание лабораторной работы**

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.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
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».
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
   1. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь.
   2. 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. Создайте следующие объекты и выведите о них информацию в консоль (N - номер Вашего варианта по списку группы):
    1. Прямоугольник синего цвета шириной N и высотой N.
    2. Круг зеленого цвета радиусом N.
    3. Квадрат красного цвета со стороной N.
    4. Также вызовите один из методов внешнего пакета, установленного с использованием pip.

**Текст программы**

**Main.py**

from lab\_python\_oop.Rectangle import Rectangle   
from lab\_python\_oop.Circle import Circle   
from lab\_python\_oop.Square import Square   
  
  
  
def main():   
print("ИУ5-51Б Гапчук Л.Д. Лаб2")   
  
rectangle = Rectangle("синего", 5, 2)   
circle = Circle("зеленого", 3)   
square = Square("красного", 4)   
  
print(rectangle)   
print(circle)   
print(square)   
  
  
if \_\_name\_\_ == "\_\_main\_\_":   
main()

**GeometricFigure.py**

from abc import ABC, abstractmethod   
  
  
class GeometricFigure(ABC):   
FIGURE\_TYPE = None   
  
@classmethod   
def get\_figure\_type(cls):   
return cls.FIGURE\_TYPE   
  
@abstractmethod   
def \_square(self):   
pass

**FigureColor.py**

class FigureColor:   
def \_\_init\_\_(self):   
self.\_color = None   
  
@property   
def color\_property(self):   
return self.\_color   
  
@color\_property.setter   
def color\_property(self, value):   
self.\_color = value

**Rectangle.py**

from lab\_python\_oop.GeometricFigure import GeometricFigure   
from lab\_python\_oop.FigureColor import FigureColor   
  
  
class Rectangle(GeometricFigure):   
FIGURE\_TYPE = "Прямоугольник"   
  
def \_\_init\_\_(self, color, width, height):   
self.\_width = width   
self.\_height = height   
self.\_figure\_color = FigureColor()   
self.\_figure\_color.color\_property = color   
  
def \_square(self):   
return self.\_width \* self.\_height   
  
def \_\_repr\_\_(self):   
return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(   
Rectangle.get\_figure\_type(),   
self.\_figure\_color.color\_property,   
self.\_width,   
self.\_height,   
self.\_square()   
)

**Circle.py**

from lab\_python\_oop.GeometricFigure import GeometricFigure   
from lab\_python\_oop.FigureColor import FigureColor   
import math   
  
  
class Circle(GeometricFigure):   
FIGURE\_TYPE = "Круг"   
  
def \_\_init\_\_(self, color, radius):   
self.radius = radius   
self.figure\_color = FigureColor()   
self.figure\_color.color\_property = color   
  
def \_square(self):   
return math.pi \* (self.radius \*\* 2)   
  
def \_\_repr\_\_(self):   
return '{} {} цвета радиусом {} площадью {}.'.format(   
Circle.get\_figure\_type(),   
self.figure\_color.color\_property,   
self.radius,   
self.\_square()   
)

**Square.py**

from lab\_python\_oop.Rectangle import Rectangle   
  
  
class Square(Rectangle):   
FIGURE\_TYPE = "Квадрат"   
  
def \_\_init\_\_(self, color, side):   
self.\_side = side   
super().\_\_init\_\_(color, self.\_side, self.\_side)   
  
def \_\_repr\_\_(self):   
return '{} {} цвета со стороной {} площадью {}.'.format(   
Square.get\_figure\_type(),   
self.\_figure\_color.color\_property,   
self.\_side,   
self.\_square()   
)

**Примеры работы программы**

