**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет "Информатика и системы управления"

Кафедра ИУ5 "Системы обработки информации и управления"

Разработка интернет-приложений

Отчет по лабораторной работе №2

Выполнила: Пасатюк Александра, группа ИУ5-53Б

Москва, 2020

**Задание**

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
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». Для вычисления площади используется константа math.pi из модуля math.
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
   * 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):
    * Прямоугольник синего цвета шириной N и высотой N.
    * Круг зеленого цвета радиусом N.
    * Квадрат красного цвета со стороной N.

**Текст программы**

Класс geom\_figur

from abc import ABC, abstractmethod

class geom\_figur(ABC):

@abstractmethod

def square(self):

pass

Класс rectangle

from lab\_pyton\_oop.geom\_figur import geom\_figur

from lab\_pyton\_oop.color import color

class rectangle(geom\_figur):

square = 0

w = 0

h = 0

FIGURE\_TYPE = "Прямоугольник"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, color\_param, w\_param, h\_param):

self.w = w\_param

self.h = h\_param

self.colour= color(color\_param)

def square(self):

return self.w\*self.h

def \_\_repr\_\_(self):

return '{} {} цвета шириной {} , высотой {} и площадью {}.'.format(

rectangle.get\_figure\_type(),

self.colour.color,

self.w,

self.h,

self.square()

)

Класс circle

from lab\_pyton\_oop.geom\_figur import geom\_figur

from lab\_pyton\_oop.color import color

import math

class circle(geom\_figur):

r = 0

\_square\_ = 0

FIGURE\_TYPE = "Круг"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

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

self.r = r

self.colour = color(color\_param)

def square(self):

return float(math.pi) \* self.r \* self.r

def \_\_repr\_\_(self):

return '{} {} цвета радиусом {} и площадью {}.'.format(

circle.get\_figure\_type(),

self.colour.color,

self.r,

self.square()

)

Класс square

from lab\_pyton\_oop.rectangle import rectangle

class square(rectangle):

FIGURE\_TYPE = "Квадрат"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

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

self.side = s

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

def \_\_repr\_\_(self):

return '{} {} цвета со стороной {} площадью {}.'.format(

square.get\_figure\_type(),

self.colour.color,

self.side,

self.square()

)

Класс color

class color:

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

self.\_color\_ = value

@property

def color(self):

return self.\_color\_

@color.setter

def color(self, value):

self.\_color\_ = value

Файл main

from lab\_pyton\_oop.rectangle import rectangle

from lab\_pyton\_oop.circle import circle

from lab\_pyton\_oop.square import square

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

r = rectangle("синего", 17, 17)

c = circle("зеленого", 17)

s = square("красного", 17)

print(r.\_\_repr\_\_())

print(c.\_\_repr\_\_())

print(s.\_\_repr\_\_())

**Результат работы программы**







