**Московский государственный технический университет им. Н.Э. Баумана**

**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)**

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №2

Выполнил:

студент группы ИУ5Ц-72Б Гусев С.Р.

Преподаватель:

Гапанюк Ю.Е

2020

**Описание задания:** разработать программу для решения биквадратного уравнения.

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:**

модуль main.py

# Лабороторная 2

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.circle import Circle

from lab\_python\_oop.square import Square

from lab\_python\_oop.figure\_color import FigureColor

#Пишем фигуры и даём им раазмер

def main():

rectangle1 = Rectangle(16, 9, "blue")

circle1 = Circle(5, "green")

square1 = Square(4, "red")

print(rectangle1)

print(circle1)

print(square1)

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

main()

модуль geometric\_figure.py

from abc import abstractmethod, ABC

#Абстрактный метод классов для дальнейшего использования и написания классов геом. фигур

class GeometricFigure(ABC):

@abstractmethod

def area(self): #self – это ссылка на текущий экземпляр класса

pass

модуль rectangle.py

from lab\_python\_oop.geometric\_figure import GeometricFigure

from lab\_python\_oop.figure\_color import FigureColor

#Класс фигур Прямоугольник

class Rectangle(GeometricFigure):

def \_\_init\_\_(self, w, h, c):

self.width = w

self.height = h

self.color = FigureColor(c)

self.type = "Прямоугольник"

def get\_type(self):

return self.type

def area(self):

return self.height \* self.width

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

return '{} с радиусом {}, который имеет площадь {} и цветом {}'.format(self.get\_type(), self.width,

self.height,

self.color, self.area())

модуль square.py

from lab\_python\_oop.rectangle import Rectangle

#Класс фигур Квадрат

class Square(Rectangle):

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

super().\_\_init\_\_(s, s, c)

self.type = "Квадрат"

def get\_type(self):

return self.type

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

return '{} с радиусом {} и цветом {}, который имеет площадь {}'.format(self.get\_type(), self.height,

self.color, self.area())

модуль circle.py

from math import pi

from lab\_python\_oop.geometric\_figure import GeometricFigure

from lab\_python\_oop.figure\_color import FigureColor

#Класс фигур Круг

class Circle(GeometricFigure):

def \_\_init\_\_(self, r, c):

self.radius = r

self.color = FigureColor(c)

self.type = "Круг"

def get\_type(self):

return self.type

def area(self):

return round(self.radius \* self.radius \* pi, 5)

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

return '{} с радиусом {} и цветом {}, который имеет площадь {}'.format(self.get\_type(), self.radius,

self.color, self.area())

модуль figure\_color.py

#Класс цвета фигур

class FigureColor:

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

self.\_color = c

def get\_color(self):

return self.\_color

def set\_color(self, c):

self.\_color = c

def del\_color(self):

del self.\_color

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

return self.\_color

color = property(get\_color, set\_color, del\_color, "Цвет фигуры 'color'")

**Окно терминала:**

