**Московский государственный технический**

**Университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2

## «Объектно-ориентированные возможности языка Python»

Выполнил:  
Власов Александр,

ИУ5-33Б

Проверил:

Гапанюк Е.Ю.

2023 г.

**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 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»:
10. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
11. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
12. 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):
13. Прямоугольник синего цвета шириной N и высотой N.
14. Круг зеленого цвета радиусом N.
15. Квадрат красного цвета со стороной N.
16. Также вызовите один из методов внешнего пакета, установленного с использованием pip.

**Текст программы**

**Main.py**

from lab\_python\_oop import abstract , circle , square , rectangle

def main():

circle\_ = circle.circle(20 , 'Красный')

circle\_.repr()

kvadrat\_ = square.kvadrat(30 , 'Зеленый')

kvadrat\_.repr()

rectangle\_ = rectangle.rectangle(10, 40 , 'Синий')

rectangle\_.repr()

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

main()

**Abstract.py**

import abc

class figure(abc.ABC):

@abc.abstractmethod

def square(self):

pass

@abc.abstractmethod

def repr(self):

pass

@abc.abstractmethod

def get\_name(self):

pass

**Circle.py**

from lab\_python\_oop import abstract, colour

from math import pi

class circle(abstract.figure):

def \_\_init\_\_(self, rad=10, colour\_="Белый "):

self.\_\_name = "Круг"

self.\_\_colour = colour.colour(colour\_)

self.\_\_rad = rad

def square(self):

res = pi \* self.\_\_rad\*\*2

return res

def get\_name(self):

return self.\_\_name

def get\_colour(self):

return self.\_\_colour

def repr(self):

print(

f"Название фигуры: {self.get\_name()}\n"

f"Цвет фигуры: {self.get\_colour()}\n"

f"Площадь фигуры: {self.square()}\n" + "\n"

)

**Rectangle.py**

from lab\_python\_oop import abstract, colour

class rectangle(abstract.figure):

def \_\_init\_\_(self, height=0, length=0, colour\_="Белый"):

self.\_length = length

self.\_\_height = height

self.\_colour = colour.colour(colour\_)

self.\_name = "Прямоугольник"

def square(self):

res = self.\_\_height \* self.\_length

return res

def get\_name(self):

return self.\_name

def get\_colour(self):

return self.\_colour

def repr(self):

print(

f"Название фигуры: {self.get\_name()}\n"

f"Цвет фигуры: {self.get\_colour()}\n"

f"Площадь фигуры: {self.square()}\n" + "\n"

)

**Square.py**

from lab\_python\_oop import abstract, colour, rectangle

class kvadrat(rectangle.rectangle):

def \_\_init\_\_(self, length=10, colour\_="Белый"):

rectangle.rectangle.\_\_init\_\_(self, length=length, colour\_=colour\_)

self.\_\_name = "Квадрат"

def square(self):

res = self.\_length\*\*2

return res

def get\_name(self):

return self.\_\_name

def get\_colour(self):

return self.\_colour

def repr(self):

print(

f"Название фигуры: {self.get\_name()}\n"

f"Цвет фигуры: {self.get\_colour()}\n"

f"Площадь фигуры: {self.square()}\n" + "\n"

)

**Colour.py**

class colour:

def \_\_init\_\_(self, colour\_="Белый"):

self.colour = colour\_

def \_\_str\_\_(self):

return self.colour

**Примеры выполнения программы**

