Отчёт по лабораторной работе №2  
по курсу «Разработка Интернет-приложений»  
Тема: «Python. Объектно-ориентированные возможности»

Выполнила:  
студентка группы ИУ5-51  
Лебедева Дарья  
Дата: \_\_\_\_\_\_ Подпись: \_\_\_\_\_\_\_  
Проверил:  
Дата: \_\_\_\_\_\_ Подпись: \_\_\_\_\_\_\_

Москва, 2018

Задание

Необходимо разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке Python 3.

2. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop. 3. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.

* 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>
* 2. 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
* 3. 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.
* 4. 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>
* 5. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.
* 6. Для классов «Прямоугольник», «Квадрат», «Круг»:

1. Определите метод "repr", который возвращает в виде строки основные  
   параметры фигуры, ее цвет и площадь. Используйте метод format https://pyformat.info/
2. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
3. 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: 1. Прямоугольник синего цвета шириной 3 и высотой 2. 2. Круг зеленого цвета радиусом 5. 3. Квадрат красного цвета со стороной 5.

Исходный код

main.py

from lab2.lab\_python\_oop.shapes import Rectangle, Circle, Square  
from lab2.lab\_python\_oop.color import Color  
def main():  
rectangle = Rectangle( 3, 2, Color( 'синий' ))  
circle = Circle( 5, Color( 'зеленый' ))  
square = Square( 5, Color( 'красный' ))  
shapes = [rectangle, circle, square]  
result = ', ' .join( str(shape) for shape in shapes)  
print( "У меня есть:" , result)  
if \_\_name\_\_ == '\_\_main\_\_' :  
# execute only if run as a script  
main()

color.py  
class Color:  
def \_\_init\_\_( self, color: str):  
self.\_color = color  
@property  
def color( self) -> str:  
return self.\_color  
def \_\_str\_\_( self) -> str:  
return self.\_color

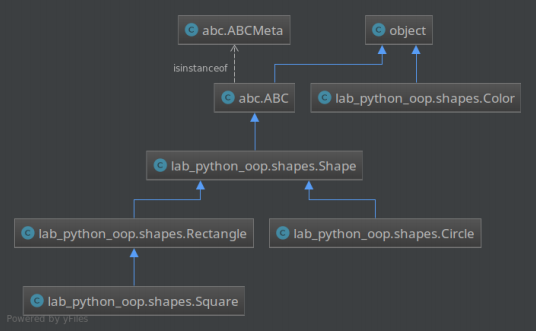
shapes.py

from abc import ABC, abstractmethod  
from .color import Color  
import math  
class Shape(ABC):  
def \_\_init\_\_( self, color: Color):  
self.\_color = color  
@abstractmethod  
def area( self) -> float:  
pass  
@property  
@abstractmethod  
def name( self) -> str:  
pass  
@abstractmethod  
def get\_data( self) -> str:  
pass  
def \_\_str\_\_( self) -> str:  
return '{} {} ({})' .format( self.\_color, self.name,  
self.get\_data())  
class Rectangle(Shape):  
def \_\_init\_\_( self, width: float, height: float, color: Color):  
self.\_width = width  
self.\_height = height  
Shape. \_\_init\_\_( self, color)  
def area( self) -> float:  
return self.\_width \* self.\_height  
@property  
def name( self) -> str:  
return 'Прямоугольник'  
def get\_data( self) -> str:  
return '{}x{}={}' .format( self.\_width, self.\_height,  
self.area())  
class Square(Rectangle):  
def \_\_init\_\_( self, side: float, color: Color):  
Rectangle. \_\_init\_\_( self, side, side, color)  
@property  
def name( self) -> str:  
return 'Квадрат'  
class Circle(Shape):  
def \_\_init\_\_( self, radius: float, color: Color):  
self.\_radius = radius  
self.\_color = color  
Shape. \_\_init\_\_( self, color)  
def area( self) -> float:  
return math.pi \* ( self.\_radius \*\* 2)  
@property  
def name( self) -> str:  
return 'Круг'  
def get\_data( self) -> str:  
return 'pi\*{}^2={}' .format( self.\_radius, self.area())

**Результаты выполнения программы:**



**Диаграмма классов:**

****