**Московский Государственный Технический Университет имени Н. Э. Баумана**



**Разработка интернет-приложений**

**Лабораторная работа №2**

ИСПОЛНИТЕЛЬ:

Зорин А.А.

ИУ5-54

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

**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.
   1. 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.

1. 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 lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.square import Square  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 rectangle = Rectangle(3, 2, "синий")  
 circle = Circle(5, "зеленый")  
 square = Square(5, "красный")  
 print(rectangle)  
 print(circle)  
 print(square)

Geometricfigure.py

from abc import ABC, abstractmethod  
  
  
class GeometricFigure(ABC):  
 @abstractmethod  
 def area(self):  
 pass  
  
 @abstractmethod  
 def get\_name(self):  
 pass

Rectangle.py

from lab\_python\_oop.geometricfigure import GeometricFigure  
from lab\_python\_oop.color import Color  
  
  
class Rectangle(GeometricFigure):  
 def \_\_init\_\_(self, width, height, figure\_color):  
 self.width = width  
 self.height = height  
 self.color = Color(figure\_color)  
 self.name = "Прямоугольник"  
  
 def area(self):  
 return self.width \* self.height  
  
 def get\_name(self):  
 return self.name  
  
 def \_\_repr\_\_(self):  
 return '{}\nШирина = {}\nВысота = {}\nПлощадь фигуры = {}\nЦвет = {}\n' \  
 .format(self.name, self.width, self.height, self.area(), self.color)

Circle.py

from lab\_python\_oop.geometricfigure import GeometricFigure  
from lab\_python\_oop.color import Color  
from math import pi  
  
  
class Circle(GeometricFigure):  
 def \_\_init\_\_(self, radius, figure\_color):  
 self.radius = radius  
 self.color = Color(figure\_color)  
 self.name = "Круг"  
  
 def area(self):  
 return self.radius \* self.radius \* pi  
  
 def get\_name(self):  
 return self.name  
  
 def \_\_repr\_\_(self):  
 return '{}\nРадиус = {}\nПлощадь фигуры = {}\nЦвет = {}\n' \  
 .format(self.name, self.radius, self.area(), self.color)

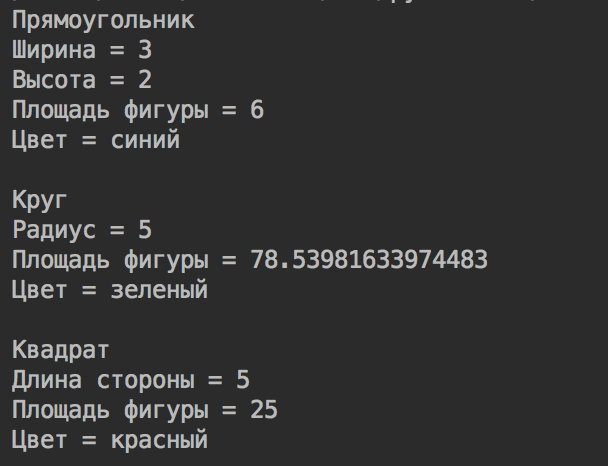
Square.py

from lab\_python\_oop.rectangle import Rectangle  
  
  
class Square(Rectangle):  
 def \_\_init\_\_(self, a, figure\_color):  
 super().\_\_init\_\_(a, a, figure\_color)  
 self.name = "Квадрат"  
  
 def \_\_repr\_\_(self):  
 return '{}\nДлина стороны = {}\nПлощадь фигуры = {}\nЦвет = {}\n' \  
 .format(self.name, self.width, self.area(), self.color)

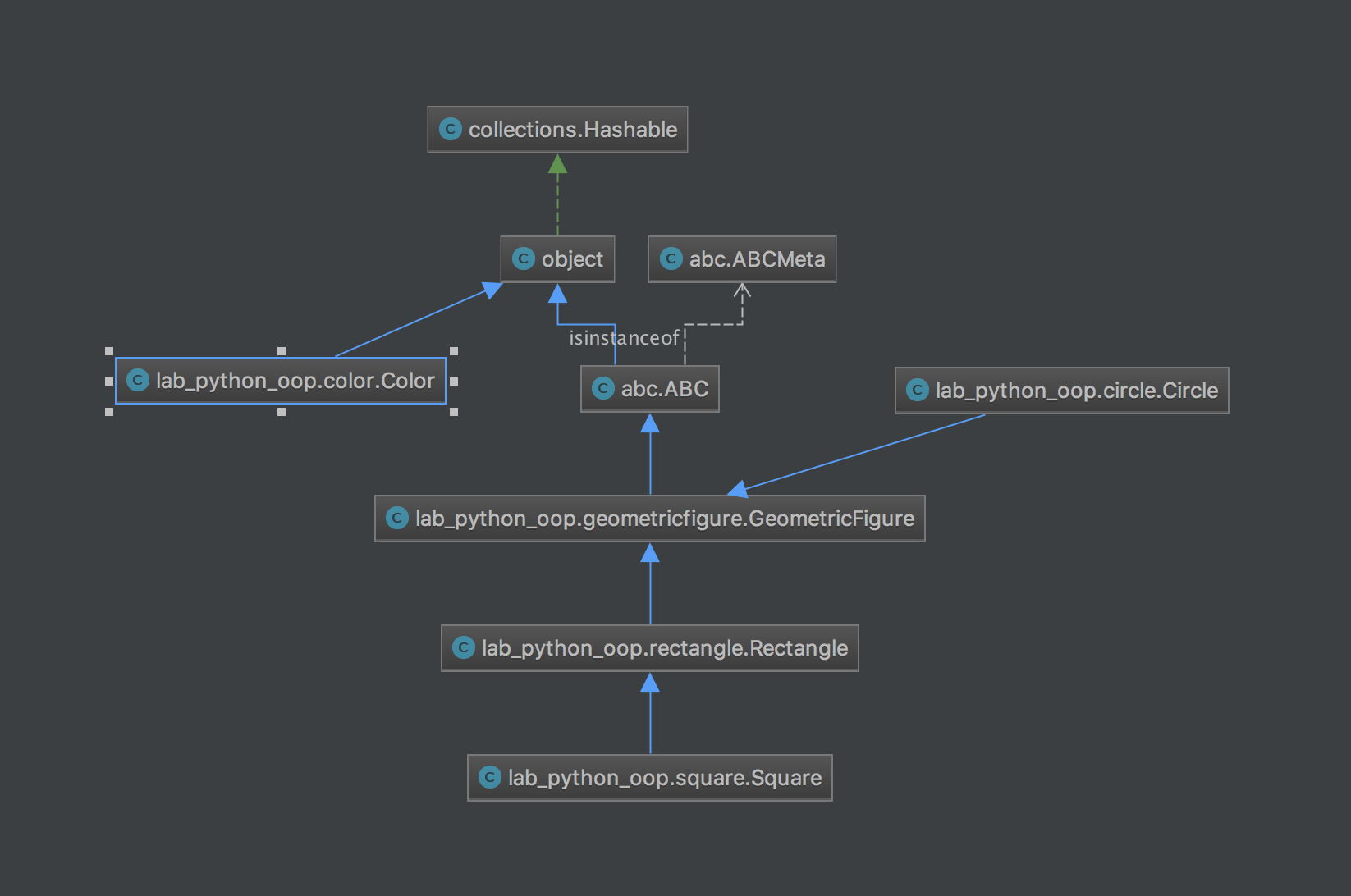
Color.py

class Color:  
 def \_\_init\_\_(self, figure\_color):  
 self.\_x = figure\_color  
  
 def get\_x(self):  
 return self.\_x  
  
 def set\_x(self, value):  
 self.\_x = value  
  
 def del\_x(self):  
 del self.\_x  
  
 def \_\_str\_\_(self):  
 return str(self.\_x)  
  
 x = property(get\_x, set\_x, del\_x, "I'm color.")

**Результаты выполнения**

****

**Диаграмма классов**

****