Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»



**Отчет**

**Лабораторная работа № 3**

**По курсу «Разработка интернет приложений»**

**«Python. Объекто-ориентированные возможности»**

**ИСПОЛНИТЕЛЬ:**

Зубков Андрей

Группа ИУ5-54

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2019 г.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю.Е.

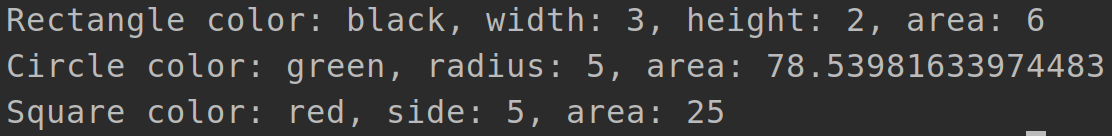
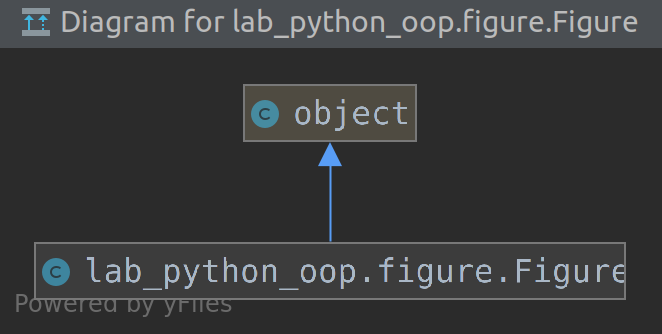
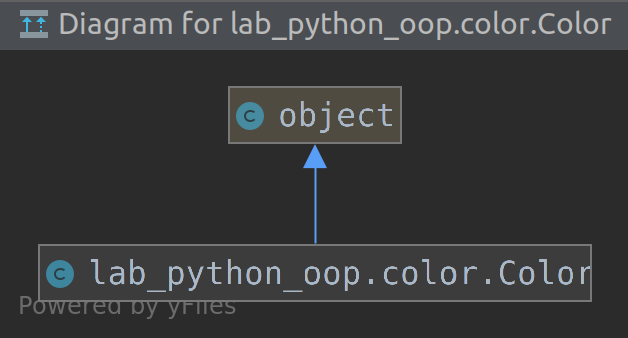
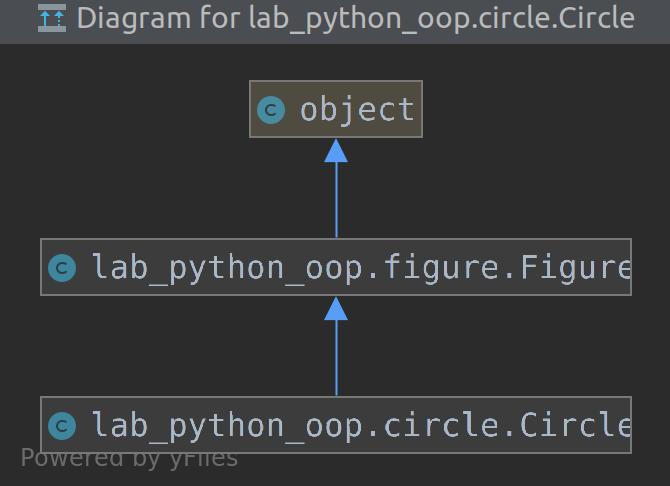
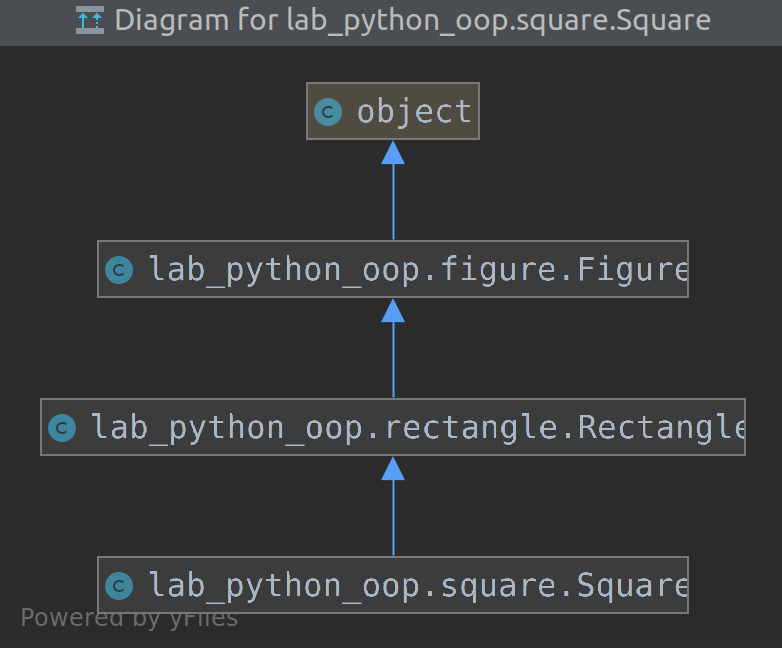
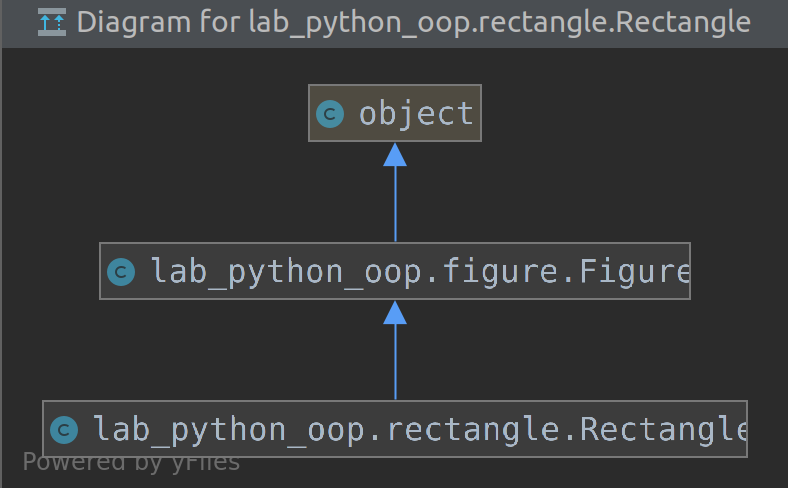
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2019 г.

Москва 2019

1. **Задание лабораторной работы**
2. 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.
   1. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры.
   2. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
   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.
   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.
      2. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
5. 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.
6. **Исходный код**

* **circle.py**from lab\_python\_oop.figure import Figure  
  from lab\_python\_oop.color import Color  
  import math  
    
  class Circle(Figure):  
    
   FIGURE\_TYPE = "Circle"  
    
   @classmethod  
   def get\_figure\_type(cls):  
   return cls.FIGURE\_TYPE  
    
   def \_\_init\_\_(self, radius, color):  
   self.radius = radius  
   self.color = Color()  
   self.color.colorproperty = color  
    
   def area(self):  
   return math.pi \* (self.radius \*\* 2)  
    
   def \_\_repr\_\_(self):  
   return "{} color: {}, radius: {}, area: {}".format(  
   Circle.get\_figure\_type(),  
   self.color.colorproperty,  
   self.radius,  
   self.area()  
   )
* **color.py**class Color:  
   def \_\_init\_\_(self):  
   self.\_\_color = None  
    
   @property  
   def colorproperty(self):  
   return self.\_\_color  
    
   @colorproperty.setter  
   def colorproperty(self, value):  
   self.\_\_color = value
* **figure.py**from abc import ABC, abstractmethod  
    
  class Figure():  
   @abstractmethod  
   def area(self):  
   pass
* **rectangle.py**from lab\_python\_oop.color import Color  
  from lab\_python\_oop.figure import Figure  
    
  class Rectangle(Figure):  
    
   FIGURE\_TYPE = "Rectangle"  
    
   @classmethod  
   def get\_figure\_type(cls):  
   return cls.FIGURE\_TYPE  
    
   def \_\_init\_\_(self, width, height, color):  
   self.width = width  
   self.height = height  
   self.color = Color()  
   self.color.colorproperty = color  
    
   def area(self):  
   return self.width \* self.height  
    
   def \_\_repr\_\_(self):  
   return "{} color: {}, width: {}, height: {}, area: {}".format(  
   Rectangle.get\_figure\_type(),  
   self.color.colorproperty,  
   self.width,  
   self.height,  
   self.area()  
   )
* **square.py**from lab\_python\_oop.rectangle import Rectangle  
    
  class Square(Rectangle):  
    
   FIGURE\_TYPE = "Square"  
    
   @classmethod  
   def get\_figure\_type(cls):  
   return cls.FIGURE\_TYPE  
    
   def \_\_init\_\_(self, side\_size, color):  
   self.side = side\_size  
   super().\_\_init\_\_(self.side, self.side, color)  
    
   def \_\_repr\_\_(self):  
   return "{} color: {}, side: {}, area: {}".format(  
   Square.get\_figure\_type(),  
   self.color.colorproperty,  
   self.side,  
   self.area()  
   )
* **main.py**from lab\_python\_oop.rectangle import Rectangle  
  from lab\_python\_oop.circle import Circle  
  from lab\_python\_oop.square import Square  
    
  def main():  
   rectangle = Rectangle(3, 2, "black")  
   circle = Circle(5, "green")  
   square = Square(5, "red")  
   print(rectangle)  
   print(circle)  
   print(square)  
    
  if \_\_name\_\_ == "\_\_main\_\_":  
   main()

1. **Скриншот работы программы**
2. **Диаграммы классов**