

МГТУ им. Н. Э. Баумана, кафедра ИУ5
курс “Разработка интернет-приложений”

Лабораторная работа №3
Python. Объектно-ориентированные
ВОЗМОЖНОСТИ

ВЫПОЛНИЛ:

Матюнин да Вейга Р.А.

Группа: ИУ5-51Б

ПРОВЕРИЛ:

Гапанюк Ю.Е.

Москва 2019

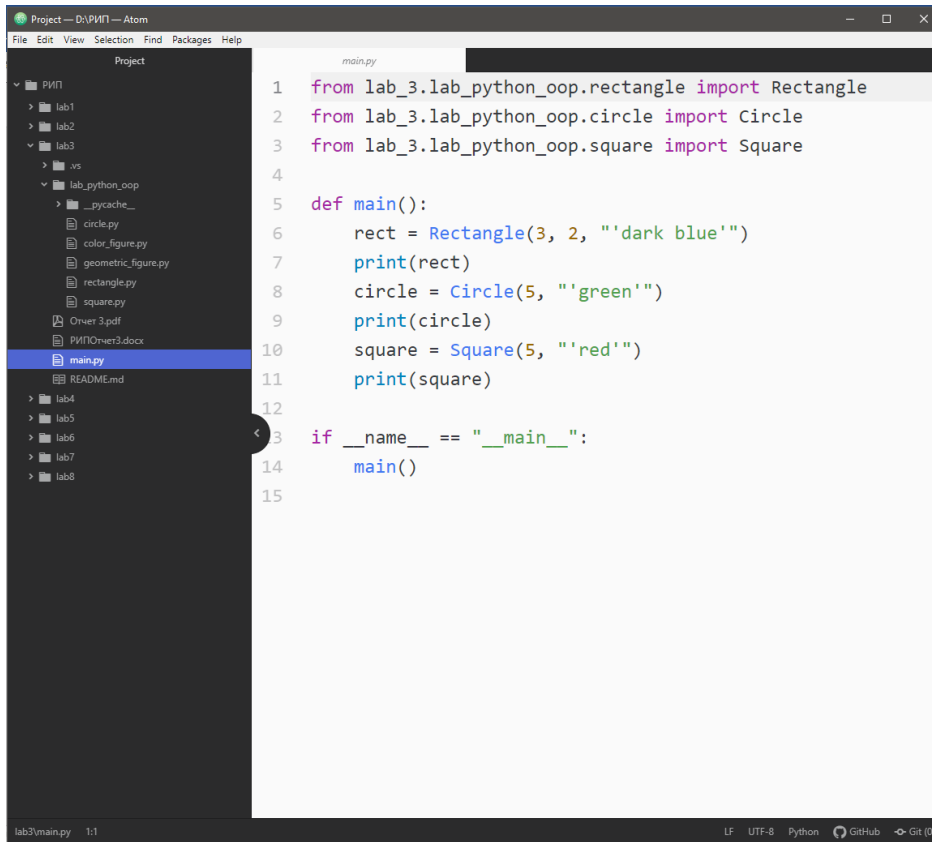
Задание и порядок выполнения

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.
 - 3.1. 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>
 - 3.2. 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>
 - 3.3. 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.
 - 3.4. 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>
 - 3.5. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.
 - 3.6. Для классов «Прямоугольник», «Квадрат», «Круг»:
 - 3.6.1. Определите метод "get", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
 - 3.6.2. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
4. 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:
 - 4.1. Прямоугольник синего цвета шириной 3 и высотой
 - 4.2. Круг зеленого цвета радиусом 5.
 - 4.3. Квадрат красного цвета со стороной 5.

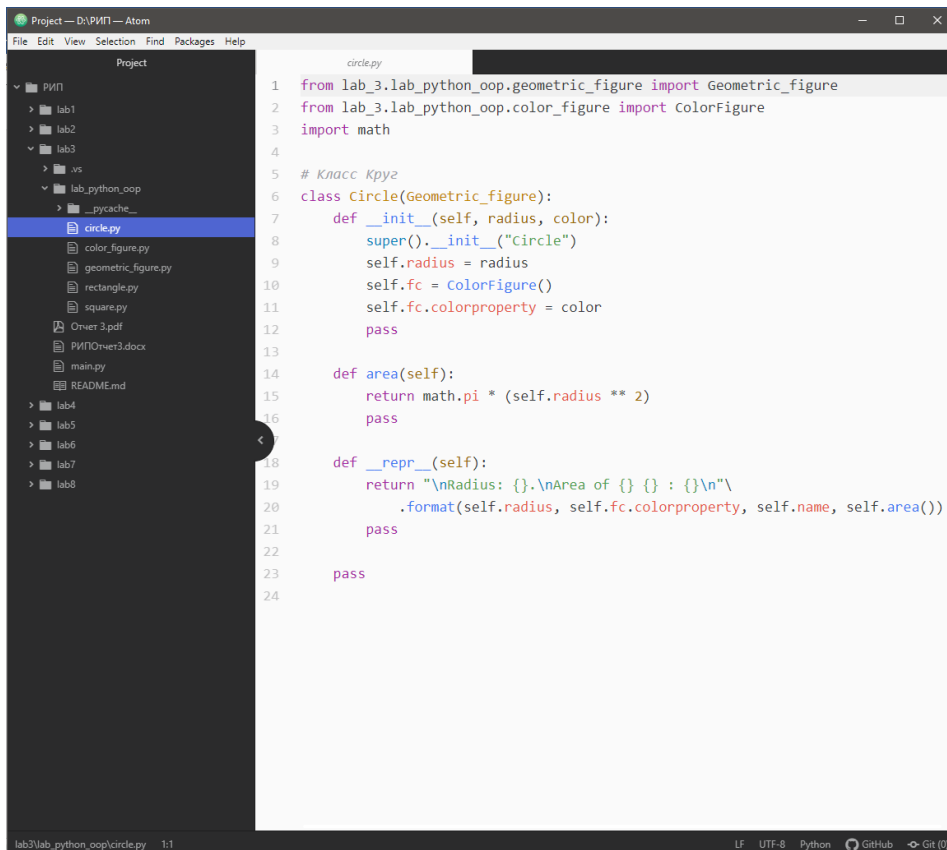
Выполненная работа

main.py



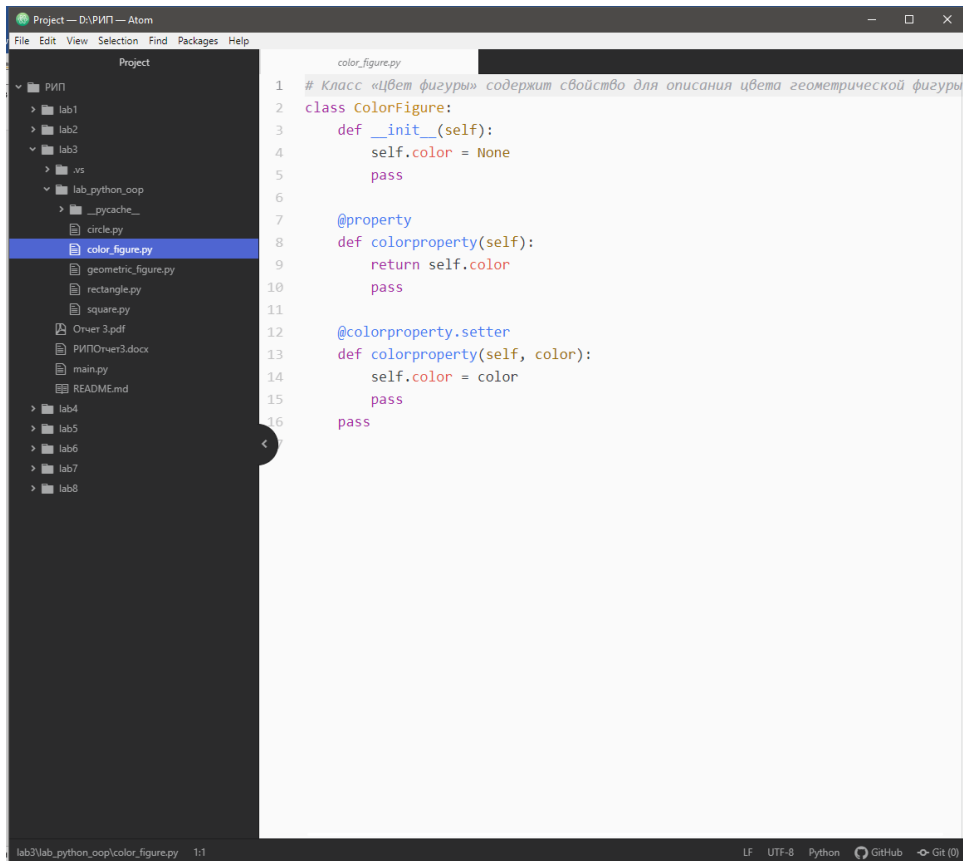
```
1 from lab_3.lab_python_oop.rectangle import Rectangle
2 from lab_3.lab_python_oop.circle import Circle
3 from lab_3.lab_python_oop.square import Square
4
5 def main():
6     rect = Rectangle(3, 2, "dark blue")
7     print(rect)
8     circle = Circle(5, "green")
9     print(circle)
10    square = Square(5, "red")
11    print(square)
12
13 if __name__ == "__main__":
14     main()
15
```

circle.py



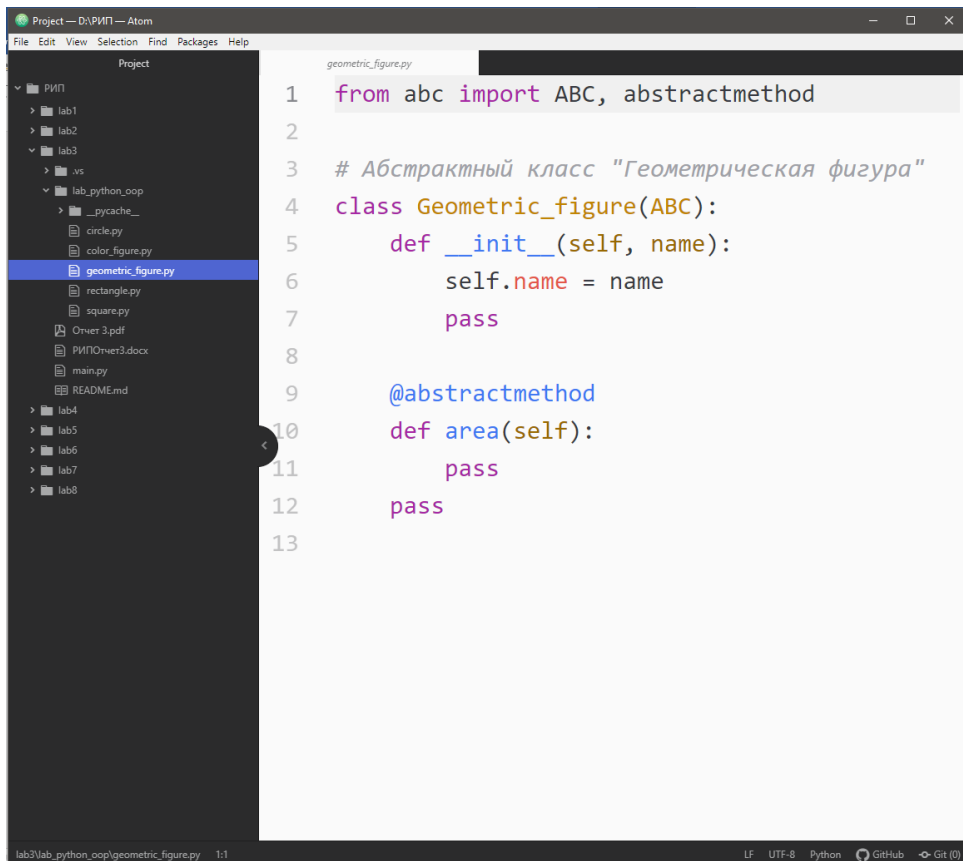
```
1 from lab_3.lab_python_oop.geometric_figure import Geometric_figure
2 from lab_3.lab_python_oop.color_figure import ColorFigure
3 import math
4
5 # Класс Круг
6 class Circle(Geometric_figure):
7     def __init__(self, radius, color):
8         super().__init__("Circle")
9         self.radius = radius
10        self.fc = ColorFigure()
11        self.fc.colorproperty = color
12        pass
13
14    def area(self):
15        return math.pi * (self.radius ** 2)
16        pass
17
18    def __repr__(self):
19        return "\nRadius: {}.\nArea of {} {} : {}\n"
20        .format(self.radius, self.fc.colorproperty, self.name, self.area())
21        pass
22
23    pass
24
```

color_figure.py



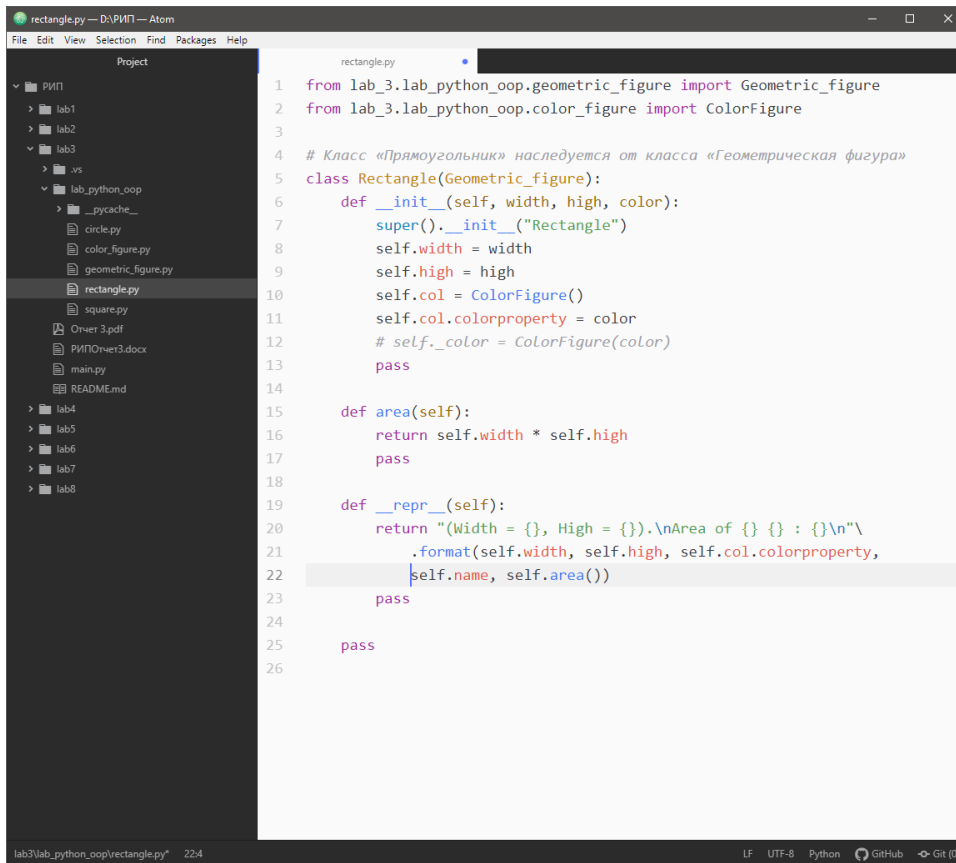
```
1 # Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры
2 class ColorFigure:
3     def __init__(self):
4         self.color = None
5         pass
6
7     @property
8     def colorproperty(self):
9         return self.color
10        pass
11
12    @colorproperty.setter
13    def colorproperty(self, color):
14        self.color = color
15        pass
16        pass
```

geometric_figure.py



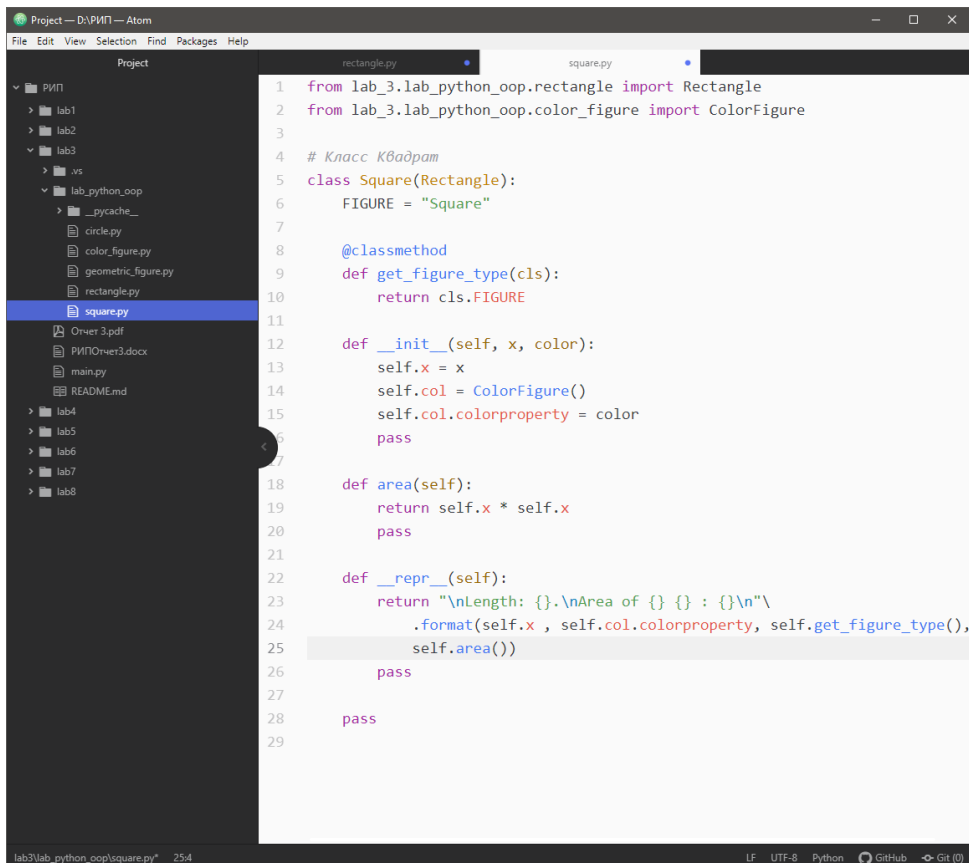
```
1 from abc import ABC, abstractmethod
2
3 # Абстрактный класс "Геометрическая фигура"
4 class Geometric_figure(ABC):
5     def __init__(self, name):
6         self.name = name
7         pass
8
9     @abstractmethod
10    def area(self):
11        pass
12    pass
13
```

rectangle.py



```
1 from lab_3.lab_python_oop.geometric_figure import Geometric_figure
2 from lab_3.lab_python_oop.color_figure import ColorFigure
3
4 # Класс «Прямоугольник» наследуется от класса «Геометрическая фигура»
5 class Rectangle(Geometric_figure):
6     def __init__(self, width, high, color):
7         super().__init__("Rectangle")
8         self.width = width
9         self.high = high
10        self.col = ColorFigure()
11        self.col.colorproperty = color
12        # self._color = ColorFigure(color)
13        pass
14
15    def area(self):
16        return self.width * self.high
17        pass
18
19    def __repr__(self):
20        return "(Width = {}, High = {}).\nArea of {} {} : {}{}\n"
21        .format(self.width, self.high, self.col.colorproperty,
22        self.name, self.area())
23        pass
24
25    pass
26
```

square.py



```
1 from lab_3.lab_python_oop.rectangle import Rectangle
2 from lab_3.lab_python_oop.color_figure import ColorFigure
3
4 # Класс Квадрат
5 class Square(Rectangle):
6     FIGURE = "Square"
7
8     @classmethod
9     def get_figure_type(cls):
10        return cls.FIGURE
11
12    def __init__(self, x, color):
13        self.x = x
14        self.col = ColorFigure()
15        self.col.colorproperty = color
16        pass
17
18    def area(self):
19        return self.x * self.x
20        pass
21
22    def __repr__(self):
23        return "\nLength: {}. \nArea of {} {} : {}{}\n"
24        .format(self.x, self.col.colorproperty, self.get_figure_type(),
25        self.area())
26        pass
27
28    pass
29
```

Результаты

Прямоугольник

Width: 3, Height: 2, Color: dark-blue, Area: 6

Круг

Radius: 5, Color: green, Area: 78.53981633974483

Квадрат

Width: 5, Height: 5, Color: red, Area: 25