МГТУ им. Н.Э. Баумана

Факультет «Информатика и системы управления»

ДИСЦИПЛИНА:

«Базы данных»

Отчет по лабораторной работе №3

«Python. Объектно-ориентированные возможности»

Выполнила:

Студент 3 курса

Факультет ИУ

Группа ИУ5-51Б

Мартынова Д.П.

08.01.2020

**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.
   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».
   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.

1. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов 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

def main():

r = Rectangle("синего", 3, 2)

c = Circle("зеленого", 5)

s = Square("красного", 5)

print(r)

print(c)

print(s)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**circle.py**

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.color import FigureColor

import math

class Circle(Figure):

FIGURE\_TYPE = "Круг"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, color\_param, r\_param):

self.r = r\_param

self.fc = FigureColor()

self.fc.colorproperty = color\_param

def square(self):

return math.pi\*(self.r\*\*2)

def \_\_repr\_\_(self):

return '{} {} цвета радиусом {} площадью {}.'.format(

Circle.get\_figure\_type(),

self.fc.colorproperty,

self.r,

self.square()

)

**color.py**

class FigureColor:

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(ABC):

@abstractmethod

def square(self):

pass

**rectangle.py**

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.color import FigureColor

class Rectangle(Figure):

FIGURE\_TYPE = "Прямоугольник"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, color\_param, width\_param, height\_param):

self.width = width\_param

self.height = height\_param

self.fc = FigureColor()

self.fc.colorproperty = color\_param

def square(self):

return self.width\*self.height

def \_\_repr\_\_(self):

return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(

Rectangle.get\_figure\_type(),

self.fc.colorproperty,

self.width,

self.height,

self.square()

)

**square.py**

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

FIGURE\_TYPE = "Квадрат"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, color\_param, side\_param):

self.side = side\_param

super().\_\_init\_\_(color\_param, self.side, self.side)

def \_\_repr\_\_(self):

return '{} {} цвета со стороной {} площадью {}.'.format(

Square.get\_figure\_type(),

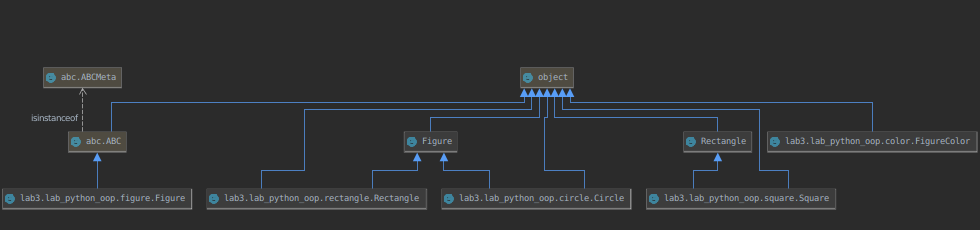
self.fc.colorproperty,

self.side,

self.square()

)

**Диаграмма классов:**



**Результат работы:**

