**Отчёт по лабораторной работе №2 по курсу «Разработка интернет приложений»**

Тема: «Python. Объектно-ориентированные возможности»

Выполнил: студент группы ИУ5-53

Андреев Алексей Андреевич

Дата: 24/09/2018 Подпись:\_\_\_\_\_\_\_

Проверил: Гапанюк Ю. Е.

Дата: \_\_\_\_\_\_ Подпись:\_\_\_\_\_\_\_

Москва, 2018

Задание лабораторной работы

**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 LabPythonOOP.circle import Circle  
from LabPythonOOP.rectangle import Rectangle  
from LabPythonOOP.square import Square  
  
if \_\_name\_\_ != "\_\_main\_\_":  
 quit(0)  
  
testrec = Rectangle(width=3, height=2, colour="Blue")  
print(testrec)  
testcir = Circle(radius=5, colour="Green")  
print(testcir)  
testsquare = Square(side=5, colour="Red")  
print(testsquare)

abstract\_colour.py:

class Colour:  
 def \_\_init\_\_(self, colour=None):  
 self.\_colour = colour  
  
 @property  
 def colour(self):  
 return self.\_colour  
  
 @colour.setter  
 def colour(self, value):  
 self.\_colour = value  
  
 @colour.deleter  
 def colour(self):  
 del self.\_colour  
  
 def \_\_repr\_\_(self):  
 return f"{self.colour}"

abstract\_figure.py:

from abc import ABC, abstractmethod  
  
  
class Figure(ABC):  
 @abstractmethod  
 def surface(self):  
 pass

circle.py:

from LabPythonOOP.abstract\_colour import Colour  
from LabPythonOOP.abstract\_figure import Figure  
  
  
class Circle(Figure):  
 def \_\_init\_\_(self, radius=0, colour="red"):  
 self.\_radius = radius  
 self.colour = Colour(colour)  
 self.name = "Круг"  
  
 def surface(self):  
 from math import pi  
 return pi \* self.\_radius\*\*2  
  
 def get\_name(self):  
 return self.name  
  
 def \_\_repr\_\_(self):  
 return f"{self.get\_name()} радиуса {self.\_radius}, с площадью {self.surface()}, цвет {self.colour}"

rectangle.py:

from LabPythonOOP.abstract\_colour import Colour  
from LabPythonOOP.abstract\_figure import Figure  
  
  
class Rectangle(Figure):  
 def \_\_init\_\_(self, width=0, height=0, colour="red"):  
 self.\_width = width  
 self.\_height = height  
 self.colour = Colour(colour)  
 self.name = "Прямоугольник"  
  
 def surface(self):  
 return self.\_width \* self.\_height  
  
 def get\_name(self):  
 return self.name  
  
 def \_\_repr\_\_(self):  
 return f"{self.get\_name()} ширины {self.\_width}, высоты {self.\_height}, с площадью {self.surface()}, цвет {self.colour}"

square.py:

from .rectangle import \*  
  
  
class Square(Rectangle):  
 def \_\_init\_\_(self, side, colour="red"):  
 super(Square, self).\_\_init\_\_(side, side, colour)  
 self.name = "Квадрат"  
  
 def get\_name(self):  
 return self.name  
  
 def \_\_repr\_\_(self):  
 return f"{self.get\_name()} со стороной {self.\_width} и площадью {self.surface()}, цвет {self.colour}"

Результаты работы:

