Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**Отчет**

**Лабораторная работа № 2**

**По курсу «Разработка интернет-приложений»**

**«Объектно-ориентированные возможности языка Python.»**

**ИСПОЛНИТЕЛЬ:**

Группа ИУ5-55

\_Кожуро Б.Е.

"20"\_октября\_2020 г.

**ПРЕПОДАВАТЕЛЬ:**

\_\_\_Гапанюк Ю.Е.\_\_

"\_\_"\_\_\_\_\_\_\_\_2020 г.

Москва 2020

1. **Задание**
   1. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.
   2. Необходимо разработать программу, реализующую работу с классами. 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.
   5. 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)
   6. 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)
   7. 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.
   8. 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)
   9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
      1. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format
      2. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
   10. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов. Создайте следующие объекты и выведите о них информацию в консоль (N - номер Вашего варианта по списку группы):
       1. Прямоугольник синего цвета шириной N и высотой N.
       2. Круг зеленого цвета радиусом N.
       3. Квадрат красного цвета со стороной N.
       4. Также вызовите один из методов внешнего пакета, установленного с использованием pip.
2. **Текст программы**
   1. **Main.py**

import numpy  
from Lab2.lab\_python\_oop.Circle import Circle  
from Lab2.lab\_python\_oop.Rectangle import Rectangle  
from Lab2.lab\_python\_oop.Square import Square  
  
# фамилия  
print("Кожуро Б. ИУ5-55Б")  
  
# проверка фигур  
  
def main():  
 r = Rectangle(8, 8, 'синего')  
 c = Circle(8, 'зеленого')  
 s = Square(8, 'красного')  
 print(r)  
 print(c)  
 print(s)  
 a = numpy.array([1, 2, 3])  
 print(type(a))  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

* 1. **FigureAbstract.py**

from abc import ABC, abstractmethod  
  
class FigureAbstract(ABC):  
 figure\_type = 'NaN'  
  
 @abstractmethod  
 def area(self):  
 pass

* 1. **FigureColour.py**

class FigureColour:  
  
 def \_\_init\_\_(self):  
 self.\_color = None  
  
 @property  
 def colourproperty(self):  
 return self.\_color  
  
 @colourproperty.setter  
 def colourproperty(self, value):  
 self.\_color = value

* 1. **Circle.py**

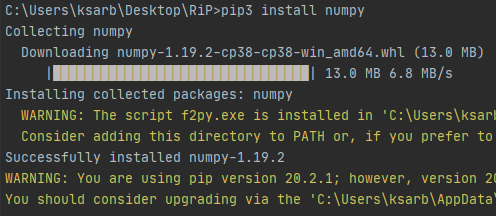
from Lab2.lab\_python\_oop.FigureAbstract import FigureAbstract  
from Lab2.lab\_python\_oop.FigureColour import FigureColour  
import math  
  
  
class Circle(FigureAbstract):  
  
 figure\_type = "Круг"  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.figure\_type  
  
 def \_\_init\_\_(self, radius, colour):  
 self.radius = radius  
 self.fc = FigureColour()  
 self.fc.colourproperty = colour  
  
 def area(self):  
 return math.pi\*(self.radius\*\*2)  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета радиусом {} площадью {}.'.format(  
 Circle.get\_figure\_type(),  
 self.fc.colourproperty,  
 self.radius,  
 self.area()  
 )

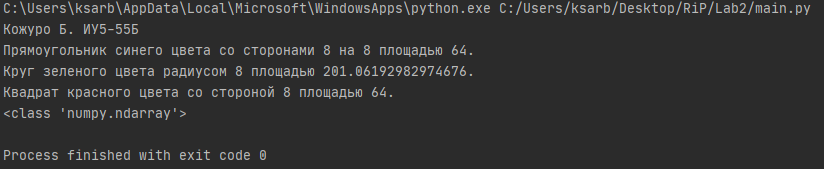
* 1. **Rectangle.py**

from Lab2.lab\_python\_oop.FigureAbstract import FigureAbstract  
from Lab2.lab\_python\_oop.FigureColour import FigureColour  
  
  
class Rectangle(FigureAbstract):  
  
 figure\_type = "Прямоугольник"  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.figure\_type  
  
 def \_\_init\_\_(self, a, b, colour):  
 self.a = a  
 self.b = b  
 self.fc = FigureColour()  
 self.fc.colourproperty = colour  
  
 def area(self):  
 return self.a \* self.b  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета со сторонами {} на {} площадью {}.'.format(  
 Rectangle.get\_figure\_type(),  
 self.fc.colourproperty,  
 self.a,  
 self.b,  
 self.area()  
 )

* 1. **Square.py**

1. from Lab2.lab\_python\_oop.FigureColour import FigureColour  
   from Lab2.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, a, colour):  
    self.a = a  
    self.fc = FigureColour()  
    self.fc.colourproperty = colour  
     
    def area(self):  
    return self.a\*\*2  
     
    def \_\_repr\_\_(self):  
    return '{} {} цвета со стороной {} площадью {}.'.format(  
    Square.get\_figure\_type(),  
    self.fc.colourproperty,  
    self.a,  
    self.area()  
    )
2. **Скриншоты выполнения**

****

****