**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Системы обработки информации и управления»

Кафедра ИУ5. Курс «РИП»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-53 |  | преподаватель каф. ИУ5 |
| Гаврилов Л. Я. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2020 г.

**Задание и порядок выполнения ЛР №2**

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»:
   * Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
   * 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 (используйте следующую конструкцию - <https://docs.python.org/3/library/__main__.html>). Создайте следующие объекты и выведите о них информацию в консоль (N - номер Вашего варианта по списку группы):
    * Прямоугольник синего цвета шириной N и высотой N.
    * Круг зеленого цвета радиусом N.
    * Квадрат красного цвета со стороной N.
    * Также вызовите один из методов внешнего пакета, установленного с использованием pip.

**Код программы**

**main.py**

from lab\_python\_oop import circle, rectangle, square

# from colorama import Fore, Back, Style, init

# Back.WHITE

# pyinstaller -F main.py

def out\_red(text):

    print("\033[31m {}" .format(text))  # красный

def out\_green(text):

    print("\033[32m {}" .format(text))  # зеленый

def out\_blue(text):

    print("\033[34m {}" .format(text))  # синий

def main():

    print(circle.Circle(5, "Зелёный"))

    print(rectangle.Rectangle(3, 2, "Синий"))

    print(square.Square(5, "Красный"))

if \_\_name\_\_ == '\_\_main\_\_':

    main()

input("Press Enter")

**circle.py**

from math import pi

from lab\_python\_oop.colour import Colour

from lab\_python\_oop.shape import Shape

class Circle(Shape):

    SHAPE\_NAME = "Круг"

    def \_\_init\_\_(self, radius, colour):

        self.radius = radius

        self.colour = Colour()

        self.colour.colour = colour

    def \_\_repr\_\_(self):

        return "{}. Цвет: {}. Радиус: {}. Площадь: {}.".format(

            Circle.get\_shape\_name(), self.colour, self.radius, self.area())

    def area(self):

        return pi \* self.radius \*\* 2

    @classmethod

    def get\_shape\_name(cls):

        return cls.SHAPE\_NAME

**colour.py**

class Colour:

    def \_\_init\_\_(self):

        self.\_colour = None

    def get\_colour(self):

        return self.\_colour

    def set\_colour(self, colour):

        self.\_colour = colour

    def del\_colour(self):

        del self.\_colour

    def \_\_repr\_\_(self):

        return str(self.\_colour)

    colour = property(get\_colour, set\_colour, del\_colour, "This is the colour property.")

**rectangle.py**

from lab\_python\_oop.colour import Colour

from lab\_python\_oop.shape import Shape

class Rectangle(Shape):

    SHAPE\_NAME = "Прямоугольник"

    def \_\_init\_\_(self, width, height, colour):

        self.width = width

        self.height = height

        self.colour = Colour()

        self.colour.colour = colour

    def \_\_repr\_\_(self):

        return "{}. Цвет: {}. Стороны: {} и {}. Площадь: {}.".format(

            Rectangle.get\_shape\_name(), self.colour, self.width, self.height, self.area())

    def area(self):

        return self.width \* self.height

    @classmethod

    def get\_shape\_name(cls):

        return cls.SHAPE\_NAME

**shape.py**

from abc import ABC, abstractmethod

class Shape(ABC):

    @abstractmethod

    def area(self):

        pass

**square.py**

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

    SHAPE\_NAME = "Квадрат"

    def \_\_init\_\_(self, side, colour):

        self.side = side

        super().\_\_init\_\_(self.side, self.side, colour)

    def \_\_repr\_\_(self):

        return "{}. Цвет: {}. Сторона: {}. Площадь: {}.".format(

            Square.get\_shape\_name(), self.colour, self.side, self.area())

    @classmethod

    def get\_shape\_name(cls):

        return cls.SHAPE\_NAME

**Скришоты выполнения**

