**Московский государственный технический университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра ИУ5 «Системы обработки информации и управления»

Отчёт по лабораторной работе №2.

“ Объектно-ориентированные возможности языка Python”

|  |  |
| --- | --- |
| Выполнил: | Проверил: |
| студент группы РТ5-31Б | преподаватель каф. ИУ5 |
| Агеев Алексей | Гапанюк Ю.Е. |
| Подпись и дата: | Подпись и дата: |

Москва, 2021 г

# Описание задания

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 - <https://pyformat.info/>
   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 (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):
    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.

# Текст программы

**Square.py**

from lab2.rectangle import Rectangle

class Square(Rectangle):

    """

    Класс «Квадрат» наследуется от класса «Прямоугольник».

    """

    FIGURE\_TYPE = "Квадрат"

    @classmethod

    def get\_figure\_type(cls):

        return cls.FIGURE\_TYPE

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

        """

        Класс должен содержать конструктор по параметрам «сторона» и «цвет».

        """

        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()

        )

**Rectangle.py**

from lab2.circle import Circle

from lab2.square import Square

from lab2.rectangle import Rectangle

def main():

    r = Rectangle("синего", 1, 1)

    c = Circle("зеленого", 1)

    s = Square("красного", 1)

    print(r)

    print(c)

    print(s)

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

    main()

**geo\_figure.py**

from abc import abstractmethod, abstractproperty

class geo\_figure:

    @abstractproperty

    def square():

        pass

**color.py**

class FigureColor:

    """

    Класс «Цвет фигуры»

    """

    def \_\_init\_\_(self):

        self.\_color = None

    @property

    def colorproperty(self):

        """

        Get-аксессор

        """

        return self.\_color

    @colorproperty.setter

    def colorproperty(self, value):

        """

        Set-аксессор

        """

        self.\_color = value

**circle.py**

from lab2.geo\_figure import geo\_figure

from lab2.color import FigureColor

import math

class Circle(geo\_figure):

    """

    Класс «Круг» наследуется от класса «Геометрическая фигура».

    """

    FIGURE\_TYPE = "Круг"

    @classmethod

    def get\_figure\_type(cls):

        return cls.FIGURE\_TYPE

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

        """

        Класс должен содержать конструктор по параметрам «радиус» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.

        """

        self.r = r\_param

        self.fc = FigureColor()

        self.fc.colorproperty = color\_param

    def square(self):

        """

        Класс должен переопределять метод, вычисляющий площадь фигуры.

        """

        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()

        )

**Main\_lab2.py**

from lab2.circle import Circle

from lab2.square import Square

from lab2.rectangle import Rectangle

def main():

    r = Rectangle("синего", 1, 1)

    c = Circle("зеленого", 1)

    s = Square("красного", 1)

    print(r)

    print(c)

    print(s)

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

    main()

**Пример работы программы.**

****