



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»
Отчет по лабораторной работе №2

Выполнила:
студент группы ИУ5-53Б
Латыпова К.Н.

Москва, 2020 г.

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](#).
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).
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`.
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
 - Определите метод `"getr"`, 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.

2. Текст программы

main.py:

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
import requests

def main():
    r = Rectangle("синего", 14, 24)
    c = Circle("зеленого", 14)
    s = Square("красного", 14)
    print(r.__repr__())
    print(c.__repr__())
    print(s.__repr__())

if __name__ == "__main__":
    main()
```

color.py:

```
class GeomFigureColor:

    def __init__(self, value):
        self._color_ = value

    @property
    def color(self):

        return self._color_

    @color.setter
    def color(self, value):

        self._color_ = value
```

circle.py:

```
from lab_python_oop.figure import GeomFigure
from lab_python_oop.color import GeomFigureColor
import math

class Circle(GeomFigure):
    _radius_ = 0
```

```

    _square_ = 0
    FIGURE_TYPE = "Круг"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color, radius,):
        self._radius_ = radius
        self.colour = GeomFigureColor(color)

    def square(self):
        return float(math.pi) * self._radius_ * self._radius_

    def __repr__(self):
        return '{} {} цвета радиусом {} и площадью {}'.format(
            Circle.get_figure_type(),
            self.colour.color,
            self._radius_,
            self.square()
        )

```

figure.py:

```

from abc import ABC, abstractmethod

class GeomFigure():
    __metaclass__ = ABC
    _square_ = 0

    @abstractmethod
    def square(self):

        pass

```

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.colour.color,
            self.side,
            self.square()
        )

```

rectangle.py:

```
from lab_python_oop.figure import GeomFigure
from lab_python_oop.color import GeomFigureColor

class Rectangle(GeomFigure):
    _square_ = 0
    _width_ = 0
    _height_ = 0
    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.colour= GeomFigureColor(color_param)
        ##self.fc = GeomFigureColor()
        ## self.fc.colorproperty = color_param

    def square(self):

        return self._width_*self._height_

    def __repr__(self):
        return '{} {} цвета шириной {} , высотой {} и площадью {}.'.format(
            Rectangle.get_figure_type(),
            self.colour.color,
            self._width_,
            self._height_,
            self.square()
        )
)
```

3. Экранные формы с примерами выполнения программы

```
Прямоугольник синего цвета шириной 14 , высотой 24 и площадью 336.
Круг зеленого цвета радиусом 14 и площадью 615.7521601035994.
Квадрат красного цвета со стороной 14 площадью 196.
```

```
Process finished with exit code 0
```