Московский государственный технический
университет 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

Выполнил:
студент группы ИУ5-53Б

Баркалова Ирина

Дата: 26.10.2021

Москва. 2021 г.

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 "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

```
import numpy as np

from lab_python_oop.circle import Circle
from lab_python_oop.rect import Rect
from lab_python_oop.square import Square

N = 3


def main():
    print(Rect(N, N, 'blue'))
    print(Circle(N, 'green'))
    print(Square(N, 'red'))
    print(np.empty(N))


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

lab_python_oop/circle.py

```
import math

from .geom_shape import GeomShape
from .shape_color import ShapeColor


class Circle(GeomShape):
    NAME = 'Круг'

    def __init__(self, radius: float, color: str):
        self._radius = radius
        self._color = ShapeColor(color)
```

```
    def square(self) -> float:
        return math.pi * self._radius * self._radius

    @classmethod
    def name(cls) -> str:
        return cls.NAME

    def __repr__(self):
        return f"<Shape: {self.NAME}, radius: {self._radius}, color:
{self._color.color_ppt}, square: {self.square()}>"
```

lab_python_oop/geom_shape.py

```
from abc import ABCMeta, abstractmethod


class GeomShape(metaclass=ABCMeta):
    @abstractmethod
    def square(self):
        pass

    @classmethod
    @abstractmethod
    def name(cls) -> str:
        pass
```

lab_python_oop/rect.py

```
from .geom_shape import GeomShape
from .shape_color import ShapeColor


class Rect(GeomShape):
    NAME = 'Прямоугольник'

    def __init__(self, height: int, width: int, color: str):
        self._height = height
        self._width = width
        self._color = ShapeColor(color)

    def square(self) -> int:
        return self._height * self._width

    @classmethod
    def name(cls) -> str:
        return cls.NAME
```

```
    def __repr__(self):
        return f"<Shape: {self.NAME}, width: {self._width}, height: {self._height},
color: {self._color.color_ppt}, " \
               f"square: {self.square()}>"
```

lab_python_oop/shape_color.py

```
class ShapeColor:
    def __init__(self, color: str):
        self._color = color

    @property
    def color_ppt(self) -> str:
        return self._color

    @color_ppt.setter
    def color_ppt(self, val: str) -> None:
        self._color = val
```

lab_python_oop/square.py

```
from .rect import Rect


class Square(Rect):
    NAME = 'Квадрат'

    def __init__(self, side: int, color: str):
        super().__init__(side, side, color)
        self._side = side

    @classmethod
    def name(cls) -> str:
        return cls.NAME

    def __repr__(self):
        return f"<Shape: {self.NAME}, side: {self._side}, color:
{self._color.color_ppt}, square: {self.square()}>"
```

Результат работы:

```
(env) iris@ThinkBook:~/PyCharmProjects/RIP/lab_2$ python main.py
<Shape: Прямоугольник, width: 3, height: 3, color: blue, square: 9>
<Shape: Круг, radius: 3, color: green, square: 28.274333882308138>
<Shape: Квадрат, side: 3, color: red, square: 9>
[1.888e-320 0.000e+000 0.000e+000]
```