

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

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

**Курс «Базовые компоненты интернет-технологий»
Отчет по лабораторной работе №2
«Объектно-ориентированные возможности языка Python»**

Выполнила:
студентка группы ИУ5-31Б
Котова Анастасия

Проверил:
Гапанюк Е.Ю.

2022 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»:
10. Определите метод `getr`, который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод `format` - <https://pyformat.info/>
11. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
12. 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):
13. Прямоугольник синего цвета шириной N и высотой N.
14. Круг зеленого цвета радиусом N.
15. Квадрат красного цвета со стороной N.
16. Также вызовите один из методов внешнего пакета, установленного с использованием `pip`.

Текст программы

Директории:

main.py

Pipfile

Pipfile.lock

lab_python_oop/

circle.py

rectangle.py

figure.py

color.py

square.py

__init__.py

main.py:

```
from lab_python_oop.rectangle import Rectangle
```

```
from lab_python_oop.circle import Circle
```

```
from lab_python_oop.square import Square
```

```
import cowsay
```

```
def main():
```

```
    r = Rectangle("синего", 3, 2)
```

```
    c = Circle("зеленого", 5)
```

```
    s = Square("красного", 5)
```

```
    print(r)
```

```
    print(c)
```

```
    print(s)
```

```
    cowsay.stegosaurus(str(r) + '\n' + str(c) + '\n' + str(s))
```

```
if __name__ == "__main__":
```

```
    main()
```

color:

```
class FigureColor:
```

```
    """
```

```
Класс «Цвет фигуры»
"""
```

```
def __init__(self):
    self._color = None
```

```
@property
def colorproperty(self):
    """
    Get-аксесор
    """
    return self._color
```

```
@colorproperty.setter
def colorproperty(self, value):
    """
    Set-аксесор
    """
    self._color = value
```

```
figure:
```

```
from abc import ABC, abstractmethod
```

```
class Figure(ABC):
    """
    Абстрактный класс «Геометрическая фигура»
    """
    @abstractmethod
    def square(self):
        """
        содержит виртуальный метод для вычисления площади фигуры.
        """
        pass
```

```
rectangle:
```

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
```

```
class Rectangle(Figure):
    """
    Класс «Прямоугольник» наследуется от класса «Геометрическая фигура».
    """
```

```
FIGURE_TYPE = "Прямоугольник"
```

```
@classmethod
```

```
def get_figure_type(cls):  
    return cls.FIGURE_TYPE
```

```
def __init__(self, color_param, width_param, height_param):  
    """
```

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

```
    """
```

```
    self.width = width_param  
    self.height = height_param  
    self.fc = FigureColor()  
    self.fc.colorproperty = color_param
```

```
def square(self):  
    """
```

Класс должен переопределять метод, вычисляющий площадь фигуры.

```
    """
```

```
    return self.width * self.height
```

```
def __repr__(self):
```

```
    return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(  
        Rectangle.get_figure_type(),  
        self.fc.colorproperty,  
        self.width,  
        self.height,  
        self.square()  
    )
```

```
circle:
```

```
from lab_python_oop.figure import Figure  
from lab_python_oop.color import FigureColor  
import math
```

```
class Circle(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()
    )
```

square:

```
from lab_python_oop.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()
    )

```

Экранные формы с примерами выполнения программы

