

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

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «БКИТ»

Отчет по лабораторной работе №2
«Объектно-ориентированные возможности языка Python»

Выполнил:

студент группы ИУ5-35Б
Трифонов Дмитрий

Подпись и дата:

Проверил:

преподаватель каф. ИУ5
Нардид А.Н.

Подпись и дата:

Москва, 2022 г.

Описание задания

Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием `pip`.

Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.

Все файлы проекта (кроме основного файла `main.py`) должны располагаться в пакете `lab_python_oop`.

Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета `lab_python_oop`.

Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь](#).

Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).

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.

Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа `math.pi` из модуля `math`.

Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:

- Определите метод "get", 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.

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`.

Дополнительное задание. Протестируйте корректность работы Вашей программы с помощью модульного теста.

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

N = 20

def main():
    r = Rectangle(N, N, "синего")
    c = Circle(N, "зеленого")
    s = Square(N, "красного")
    print(r)
    print(c)
    print(s)

    print('\n\nВызов метода из сторонней библиотеки:')
    r = requests.get('https://api.github.com/events')
    print(r.encoding)

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

Файлы пакета lab_python_oop:

circle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.figure_color import Figure_color
import math

class Circle(Figure):

    FIGURE_TYPE = 'Круг'

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

    def __init__(self, radius, color):
        self._radius = radius
        self._figure_color = Figure_color()
        self._figure_color.colorproperty = color
```

```

def get_area(self):
    return math.pi * (self._radius ** 2)

def __repr__(self):
    return '{} {} цвета радиусом {} площадью {}'.format(
        Circle.get_figure_type(),
        self._figure_color.colorproperty,
        self._radius,
        self.get_area()
    )

```

figure_color.py

```

class Figure_color:
    def __init__(self):
        self._color = None

    @property
    def colorproperty(self):
        return self._color

    @colorproperty.setter
    def colorproperty(self, value):
        self._color = value

```

figure.py

```

from abc import ABC, abstractmethod
class Figure(ABC):
    @abstractmethod
    def get_area(self):
        pass

```

rectangle.py

```

from lab_python_oop.figure import Figure
from lab_python_oop.figure_color import Figure_color
class Rectangle(Figure):
    FIGURE_TYPE = "Прямоугольник"

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

    def __init__(self, width, height, color):

```

```

        self._figure_color = Figure_color()
        self._width = width
        self._height = height
        self._figure_color.colorproperty = color

def get_area(self):
    return self._width * self._height

def __repr__(self):
    return '{} {} цвета шириной {} и высотой {} площадью {}'.format(
        Rectangle.get_figure_type(),
        self._figure_color.colorproperty,
        self._width,
        self._height,
        self.get_area()
    )

```

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

    def __repr__(self):
        return '{} {} цвета со стороной {} площадью {}'.format(
            Square.get_figure_type(),
            self._figure_color.colorproperty,
            self._side,
            self.get_area()
        )

```

Экранные формы

```
(bkit-knEZUJXe) PS C:\Users\Dmitriy\Documents\Programming\bkit\2> python main.py
Прямоугольник синего цвета шириной 20 и высотой 20 площадью 400.
Круг зеленого цвета радиусом 20 площадью 1256.6370614359173.
Квадрат красного цвета со стороной 20 площадью 400.
```

Вызов метода из сторонней библиотеки:

utf-8

```
(bkit-knEZUJXe) PS C:\Users\Dmitriy\Documents\Programming\bkit\2> █
```