



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»

Отчет по лабораторной работе № 2
Разработка интернет приложений

Студент:
группы ИУ5-54Б
Ведьгун Е.А.

2020 г.

Москва

Цель работы

изучение объектно-ориентированных возможностей языка Python.

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`.

Текст программы

Circle

```
from Lab2.lab_python_oop.geometric_figure_class import Geometric_figure
from Lab2.lab_python_oop.quadrate_class import Quadrate
from Lab2.lab_python_oop.color_class import Color
from math import pi

class Circle(Quadrate):
    figure_type = "круг"
    def __init__(self, r, color):
        self.r = r
        super().__init__(r, color)

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

    def Square(self):
        return pi*self.r*self.r

    def __repr__(self):
        return ("тип фигуры: {}\nрадиус: {}\nцвет: {}\nплощадь: {}\n\n".format(
            Circle.get_figure_type(),
            self.r,
            self.fc.color,
            self.Square()
        ))
```

color

```
class Color:
    def __init__(self):
        self.__color = None

    @property
    def color(self):
        return self.__color

    @color.setter
    def color(self, color):
        self.__color = color
```

figure

```
from abc import ABC, abstractmethod
class Geometric_figure(ABC):
    @abstractmethod
    def Square(self):
        pass
```

quadrate

```
from Lab2.lab_python_oop.rectangle_class import Rectangle
class Quadrate(Rectangle):
    figure_type = "квадрат"
    def __init__(self, side, color):
        self.side = side
        super().__init__(side, side, color)

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

    def __repr__(self):
        return ("тип фигуры: {}\nдлина: {}\nцвет: {}\nплощадь: {}\n\n".format(
            (
                Quadrate.get_figure_type(),
                self.width,
                self.fc.color,
                self.Square()
            )
        ))
```

rectangle

```
from Lab2.lab_python_oop.geometric_figure_class import Geometric_figure
from Lab2.lab_python_oop.color_class import Color
class Rectangle(Geometric_figure):
    figure_type = "Прямоугольник"

    def __init__(self, width, heigth, color):
        self.width = width
        self.heigth = heigth
        self.fc = Color()
        self.fc.color = color

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

    def Square(self):
        return self.width*self.heigth

    def __repr__(self):
        return ("тип фигуры: {}\nширина: {}\nвысота: {}\nцвет: {}\nплощадь:
{}\n\n".format
            (
                Rectangle.get_figure_type(),
                self.width,
                self.heigth,
                self.fc.color,
                self.Square()
            )
        ))
```

main

```
from Lab2.lab_python_oop.geometric_figure_class import Geometric_figure
from Lab2.lab_python_oop.color_class import Color
from Lab2.lab_python_oop.rectangle_class import Rectangle
from Lab2.lab_python_oop.quadrate_class import Quadrate
from Lab2.lab_python_oop.circle_class import Circle

def main():
    Rectangle1 = Rectangle(3, 4, "красный")
    print(Rectangle1)
    Quadrate1 = Quadrate(3, "green")
    print(Quadrate1)
    Circle1 = Circle(3, "желтый")
    print(Circle1)
if __name__ == "__main__":
    main()
```