

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

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Разработка Интернет-приложений»

Отчет по лабораторной работе №3
«Python. Объектно-ориентированные возможности»

Выполнил:

студент группы ИУ5-54Б

Коваленко Артём

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата:

Подпись и дата:

Москва, 2019 г.

Описание задания лабораторной работы

Необходимо разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке Python 3.
2. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab_python_oop.
3. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab_python_oop.
 1. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать здесь - <https://docs.python.org/3/library/abc.html>
 2. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь - <https://docs.python.org/3/library/functions.html#property>
 3. 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.
 4. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math - <https://docs.python.org/3/library/math.html>
 5. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.
 6. Для классов «Прямоугольник», «Квадрат», «Круг»:
 1. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
 2. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
4. 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:
 1. Прямоугольник синего цвета шириной 3 и высотой 2.
 2. Круг зеленого цвета радиусом 5.
 3. Квадрат красного цвета со стороной 5.

Source

- figure.py

```
from abc import ABC, abstractmethod
class GeometryFigure(ABC):
    def square(self):
        pass
    def repr(self):
        pass
```

- figcolor.py

```
class FigureColor(object):
    def __init__(self, col):
        """Constructor of figure class"""
        FigureColor._value = col
    @property
    def value(self):
        return self._value
    @value.setter
    def value(self, val):
        self._value = val
    @value.deleter
    def value(self):
        del self._value
```

- rectangle.py

```
from lab3.figure import GeometryFigure
from lab3.figcolor import FigureColor
class Rect(GeometryFigure, FigureColor):
    def __init__(self, width, height, color):
        self._w = width
        self._h = height
        self._name = "Rect"
        super().__init__(color)
    """width prop"""
    @property
    def width(self):
        return self._w
    @width.setter
    def width(self, w):
        self._w = w
    @width.deleter
    def width(self):
        del self._w
    """height prop"""
    @property
    def height(self):
        return self._h
    @height.setter
```

```

def height(self, h):
    self._h = h
@height.deleter
def height(self):
    del self._h
def square(self):
    return self._h * self._w
def repr(self):
    return "It's {3}, color {0}, width {1}, height
{2}".format(self._value, self._w, self._h, self._name)

```

- circle.py

```

from lab3.figure import GeometryFigure
from lab3.figcolor import FigureColor
import math
class Circle(GeometryFigure, FigureColor):
    def __init__(self, radius, color):
        self._r = radius
        self._name = "Circle"
        super().__init__(color)
    @property
    def radius(self):
        return self._r
    @radius.setter
    def radius(self, r):
        self._r = r
    @radius.deleter
    def radius(self):
        del self._r
    @property
    def square(self):
        return math.pi * (self._r ** 2)
    def repr(self):
        return "It's {2}, color {0}, radius
{1}".format(self._value, self._r, self._name)

```

- square.py

```

from lab3.rectangle import Rect
class Square(Rect):
    def __init__(self, a, color):
        self._name = "Square"
        super().__init__(a, a, color)
    def repr(self):
        return "It's {2}, color {0}, side length
{1}".format(self._value, self._w, self._name)

```

- __main__.py

```

from lab3.rectangle import Rect
from lab3.square import Square

```

```
from lab3.circle import Circle
def main():
    r = Rect(2, 17, "Purple")
    c = Circle(55, "Yellow")
    s = Square(-3, "Green")
    print(r.repr())
    print(c.repr())
    print(s.repr())
if __name__ == "__main__":
    main()
```

Скриншоты с результатами выполнения

```
It's Rect, color Green, width 2, height 17
It's Circle, color Green, radius 55
It's Rect, color Green, side length -3

Process finished with exit code 0
```