

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

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет лабораторной работе №2

Выполнил:

студент группы ИУ5-34Б

Баширов Гелани

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 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.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать.
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 `"get"`, который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод `format`.
 - 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. 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.

Текст программы

```
from abc import ABC, abstractmethod
import math

class Geom_Figure(ABC):
    @abstractmethod
    def square(self):
        pass

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

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

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

class Rectangle(Geom_Figure):
    figure_type = "Rectangle"

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

    def __init__(self, width, height, colour):
        self._width = width
        self._height = height
        self.fc = Figure_Colour()
        self.fc.colorproperty = colour

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

    def __repr__(self):
        return "{}".format(self.get_type(),
self._width, self._height,
self.fc.colorproperty,
self.square())

class Circle(Geom_Figure):
    figure_type = "Circle"

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

    def __init__(self, radius, colour):
        self._radius = radius
        self.fc = Figure_Colour()
        self.fc.colorproperty = colour
```

```

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

    def __repr__(self):
        return "{} , radius = {}, colour = {}, square = {}".format(self.get_type(),
                                                         self._radius,
                                                         self.fc.colorproperty,
                                                         self.square())

class Square(Rectangle):
    figure_type = "Square"

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

    def __init__(self, side, colour):
        self._side = side
        super().__init__(self._side, self._side, colour)

    def __repr__(self):
        return "{} , side = {}, colour = {}, square = {}".format(self.get_type(),
                                                         self._side,
                                                         self.fc.colorproperty,
                                                         self.square())

def main():
    r = Rectangle(3, 2, "Blue", )
    c = Circle(5, "Green")
    s = Square(5, "Red", )
    print(r)
    print(c)
    print(s)

if __name__ == "__main__":
    main()

```

Анализ результатов

```

Rectangle, width = 3, height = 2, colour = Blue, square = 6
Circle, radius = 5, colour = Green, square = 78.53981633974483
Square, side = 5, colour = Red, square = 25

```

