

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

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

Курс «Парадигмы и конструкции языков программирования»

Отчет по ЛР№2

Выполнил:
студент группы ИУ5-34Б
Кожевников М. А.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2023 г.

Задание:

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 прочесть [здесь](#).

Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочесть [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 `getr`, 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`.

Текст программы:

```
main.py  circlecl.py  colorcl.py x  figure.py  rectanglecl.py  squaresp.py
1  class colorofshape:
2      def __init__(self, color):
3          self.__color = color
4
5      @property
6      def color(self):
7          return self.__color
8
9      @color.setter
10     def color(self, color):
11         self.__color = color

main.py x  circlecl.py  colorcl.py  figure.py x  rectanglecl.py  squaresp.py
1  from abc import ABC, abstractmethod
2
3
4  class figuresh(ABC):
5      @abstractmethod
6      def __repr__(self):
7          pass
8
9      @abstractmethod
10     def square(self):
11         pass
12

main.py  circlecl.py  colorcl.py  figure.py  rectanglecl.py x  squaresp.py
1  from lab_python_oop.figure import figuresh
2  from lab_python_oop.colorcl import colorofshape
3
4
5  class rectangle(figuresh):
6      name = "rectangle"
7
8      def __init__(self, width, height, color):
9          self.__width = width
10         self.__height = height
11         self.__color = colorofshape(color).color
12
13     def square(self):
14         return self.__width * self.__height
15
16     def __repr__(self):
17         return (f"name: {self.name} | width : {self.__width} | height: {self.__height} | color : {self.__color} | square : {self.__width * self.__height}")
18
19
```

```
main.py  circlecl.py  colorcl.py  figure.py  rectanglecl.py  squaresp.py x
1  from lab_python_oop.rectanglecl import rectangle
2  from lab_python_oop.colorcl import colorofshape
3
4
5  class squareshape(rectangle):
6      name = "square"
7
8      def __init__(self, width, color):
9          self.__width = width
10         self.__color = colorofshape(color).color
11
12     def __repr__(self):
13         return (f"name: {self.name} | width : {self.__width} | color : {self.__color} | square : {self.__width ** 2}")
14
15     def square(self):
16         return self.__width ** 2
17
main.py x  circlecl.py x  colorcl.py  figure.py  rectanglecl.py  squaresp.py
1  from lab_python_oop.figure import figuresh
2  from lab_python_oop.colorcl import colorofshape
3  from math import pi
4
5
6  class circle(figuresh):
7      name = "circle"
8
9      def __init__(self, radius, color):
10         self.__radius = radius
11         self.__color = colorofshape(color).color
12
13     def __repr__(self):
14         return (
15             f"name: {self.name} | radius : {self.__radius} | color : {self.__color} | square : {int(pi ** 2 * self.__radius)}")
16
17     def square(self):
18         return pi ** 2 * self.__radius
19
main.py x  circlecl.py  colorcl.py  figure.py  rectanglecl.py  squaresp.py
1  if __name__ == "__main__":
2      from lab_python_oop.rectanglecl import rectangle
3      from lab_python_oop.circlecl import circle
4      from lab_python_oop.squaresp import squareshape
5      from termcolor import colored
6
7      rec = rectangle(8, 8, "blue")
8      cir = circle(8, "green")
9      sq = squareshape(8, "red")
10
11     print(colored(rec, "blue"))
12     print(colored(cir, "green"))
13     print(colored(sq, "red"))
14
```

Результат работы программы:

```
name: rectangle | width : 8 | height: 8 | color : blue | square : 64
name: circle | radius : 8 | color : green | square : 78
name: square | width : 8 | color : red | square : 64
```