*Государственное образовательное учреждение высшего профессионального образования*



***«***

***Московский***

***государственный***

***технический***

***университет***

***имени***

***Н***

***.***

***Э***

***.***

***Баумана***

***»***

***(***

***МГТУ***

***им***

***.***

***Н***

***.***

***Э***

***.***

***Баумана***

***)***

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Системы обработки информации и управления» (ИУ-5)

**ЛАБОРАТОРНАЯ РАБОТА**

**№ 2**

\_\_\_\_\_\_\_\_\_Объектно-ориентированные возможности языка Python\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Группа ИУ5-35Б

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **16.12.2024** /**Д.Е. Мушкарин /**

(Подпись, дата) (И.О.Фамилия)

Преподаватель \_\_\_\_\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** /**Ю. Е. Гапанюк/**

(Подпись, дата) (И.О.Фамилия)

2024

### **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 прочитать [здесь.](https://docs.python.org/3/library/abc.html)
* 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" \l "property)
* 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.](https://docs.python.org/3/library/math.html)
* 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.
* 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 (Python):**

**main.py**

from Square import Square

from Circle import Circle

from Rectangle import Rectangle

import pyfiglet # type: ignore

print(pyfiglet.figlet\_format("laba!", font="slant"))

def main():

Rectangle(13,13, "синий").repr()

Circle(13, "зеленый").repr()

Square(13, "красный").repr()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Color.py**

class Color:

def \_\_init\_\_(self, color : str):

self.color = color

@property

def color(self) -> str:

return self.\_color

@color.setter

def color(self, color : str):

self.\_color = color

**AbstractShape.py**

from abc import ABC, abstractmethod

class AbstractShape(ABC):

@abstractmethod

def area(self):

pass

**Square.py**

from Rectangle import Rectangle

class Square(Rectangle):

name = "Квадрат"

def \_\_init\_\_(self, side : float, color : str):

super().\_\_init\_\_(side, side, color)

def area(self) -> float:

return self.\_width \* self.\_height

def get\_name(cls) -> str:

return cls.name

def repr(self):

print(f"{self.get\_name()} со стороной {self.\_width}, площадью {self.area()}, {self.\_color.color} цвет")

**Rectangle.py**

from Color import Color

from AbstractShape import AbstractShape

class Rectangle(AbstractShape):

name = "Прямоугольник"

def \_\_init\_\_(self, width : float, height : float, color : str):

self.\_width = width

self.\_height = height

self.\_color = Color(color)

def area(self) -> float:

return self.\_width \* self.\_height

def get\_name(cls) -> str:

return cls.name

def repr(self):

print(f"{self.get\_name()} ширины {self.\_width}, высоты {self.\_height}, площадью {self.area()}, {self.\_color.color} цвет")

**Circle.py**

from Color import Color

from AbstractShape import AbstractShape

from math import pi

class Circle(AbstractShape):

name = "Круг"

def \_\_init\_\_(self, radius : float, color : str):

self.\_radius = radius

self.\_color = Color(color)

def area(self) -> float:

return pi\*self.\_radius\*\*2

def get\_name(cls) -> str:

return cls.name

def repr(self):

print(f"{self.get\_name()} с радиусом {self.\_radius}, площадью {self.area():.4f}, {self.\_color.color} цвет")

**Вывод:**

