**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчёт по лабораторной работе № 2.

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31б  Коломийчук Илья Валерьевич |  | преподаватель каф. ИУ5  Гапанюк Ю.Е. |
|  |  |  |
| Подпись и дата: |  | 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 "repr", 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.

1. 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

Main.py

# from lab\_python\_oop import figure

# # from lab\_python\_oop import color\_figure

# from lab\_python\_oop import square

# from lab\_python\_oop import rectangle

# from lab\_python\_oop import circle

import lab\_python\_oop as figures

def main():

    a = figures.square. Square(5,"red")

    print(a.repr())

    b = figures.rectangle.Rectangle(5, 6, "green")

    print(b.repr())

    c = figures.circle.Circle(5, "yellow")

    print(c.repr())

if (\_\_name\_\_ == '\_\_main\_\_'):

    main()

\_\_init\_\_.py

# from .circle import \*

# from .color\_figure import \*

# from .figure import \*

# from .rectangle import \*

# from .square import \*

from . import circle, color\_figure, figure, rectangle, square

circle.py

from . import figure

from . import color\_figure

from math import pi

class Circle(figure.Figure):

    def \_\_init\_\_(self, r, color):

        self.r = r

        self.color = color\_figure.Color(color)

    FIGURE\_TYPE = "Circle"

    def Get\_Square(self):

        return pi \* self.r \*\* 2

color\_figure.py

class Color():

    def \_\_init\_\_(self, name\_of\_color):

        self.color = name\_of\_color

    def Get\_Value(self):

        return self.color

figure.py

from abc import ABC

from abc import abstractmethod

from . import color\_figure

class Figure(ABC):

    FIGURE\_TYPE = ""

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

    def Get\_Square(self):

        pass

    def Get\_type\_figure(self):

        return self.FIGURE\_TYPE

    def repr(self):

        return ('{} {} {}'.format(self.Get\_type\_figure(), self.Get\_Square(), self.color.Get\_Value()))

rectangle.py

from . import figure

from . import color\_figure

class Rectangle(figure.Figure):

    def \_\_init\_\_(self, a, b, color):

        self.a = a

        self.b = b

        self.color = color\_figure.Color(color)

    FIGURE\_TYPE = "Rectangle"

    def Get\_Square(self):

        return self.a \* self.b

square.py

from . import rectangle

from . import color\_figure

class Square(rectangle.Rectangle):

    def \_\_init\_\_(self, a, color):

        self.a = a

        self.b = a

        self.color = color\_figure.Color(color)

    FIGURE\_TYPE = "Square"

    def Get\_Square(self):

        return super().Get\_Square()

**Анализ результатов**

