**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет ИУ

Кафедра ИУ5

Курс «Основы информатики»

Отчет по лабораторной работе №2

## “Объектно-ориентированные возможности языка Python.”

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы 36б:  Азиев А. А. |  | преподаватель каф. \_\_\_\_ |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024г.

**Задача:**

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 [здесь.](https://docs.python.org/3/library/abc.html)
6. 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)
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.](https://docs.python.org/3/library/math.html)
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
10. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
11. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
12. 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.

1. Также вызовите один из методов внешнего пакета, установленного с использованием pip.

**Код программы:**

oop.abstract\_figure.py

from abc import ABC, abstractmethod

class GeometricFigure(ABC):

    @abstractmethod

    def area(self):

        pass

    @abstractmethod

    def \_\_repr\_\_(self):

        pass

oop.circle.py

import math

from oop.color import Color

from oop.abstract\_figure import GeometricFigure

class Circle(GeometricFigure):

    shape\_name = "Круг"

    def \_\_init\_\_(self, radius, color\_name):

        self.radius = radius

        self.color = Color(color\_name)

    def area(self):

        return math.pi \* self.radius \*\* 2

    def \_\_repr\_\_(self):

        return "{} (радиус: {}, цвет: {}, площадь: {:.2f})".format(

            self.shape\_name, self.radius, self.color, self.area())

oop.color.py

class Color:

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

        self.color\_name = color\_name

    def \_\_repr\_\_(self):

        return self.color\_name

oop.rectangle.py

from oop.color import Color

from oop.abstract\_figure import GeometricFigure

class Rectangle(GeometricFigure):

    shape\_name = "Прямоугольник"

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

        self.width = width

        self.height = height

        self.color = Color(color\_name)

    def area(self):

        return self.width \* self.height

    def \_\_repr\_\_(self):

        return "{} (ширина: {}, высота: {}, цвет: {}, площадь: {})".format(

            self.shape\_name, self.width, self.height, self.color, self.area())

oop.square.py

from oop.rectangle import Rectangle

class Square(Rectangle):

    shape\_name = "Квадрат"

    def \_\_init\_\_(self, side\_length, color\_name):

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

    def \_\_repr\_\_(self):

        return "{} (сторона: {}, цвет: {}, площадь: {})".format(

            self.shape\_name, self.width, self.color, self.area())

main.py

from oop.rectangle import Rectangle

from oop.circle import Circle

from oop.square import Square

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

    rect = Rectangle(5, 3, "красный")

    print(rect)

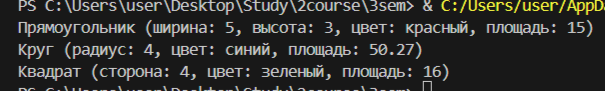
    circ = Circle(4, "синий")

    print(circ)

    sq = Square(4, "зеленый")

    print(sq)

**Результат выполнения:**

****