**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б: |  | преподаватель каф. ИУ5 |
| Семчёнок Владимир Дмитриевич |  | 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.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
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.
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.
    * Также вызовите один из методов внешнего пакета, установленного с использованием pip.

**Листинг**

*main.py*

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.circle import Circle

from lab\_python\_oop.quadrate import Quadrate

import numpy as np

def main():

rec = Rectangle(3, 3, "синий")

cir = Circle(3, "зелёный")

squ = Quadrate(3, "красный")

print(rec)

print(cir)

print(squ)

print(np.ones((4,4)))

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

main()

*circle.py*

from lab\_python\_oop.geometrical\_figure import GeometricalFigure

from lab\_python\_oop.f\_color import FigurColor

from math import pi

class Circle(GeometricalFigure):

FIGURE\_TYPE = "Круг"

@classmethod

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

self.radius = radius

self.color = FigurColor()

self.color.colorproperty = color

def get\_type(self):

return self.FIGURE\_TYPE

def get\_square(self):

return pi\*self.radius\*self.radius

def \_\_repr\_\_(self):

return '{}, цвет - {}, радиус = {}, площадь = {}.'.format(

self.get\_type(),

self.color.colorproperty,

self.radius,

self.get\_square()

)

*f\_color.py*

class FigurColor:

def \_\_init\_\_(self):

self.\_color = None

@property

def colorproperty(self):

"""

Getter

"""

return self.\_color

@colorproperty.setter

def colorproperty(self, value):

"""

Setter

"""

self.\_color = value

*geometrical\_figure.py*

from abc import ABC, abstractmethod

class GeometricalFigure(ABC):

@abstractmethod

def get\_square(self):

pass

*quadrate.py*

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.f\_color import FigurColor

class Quadrate(Rectangle):

FIGURE\_TYPE = "Квадрат"

@classmethod

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

self.side = side

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

def get\_type(self):

return self.FIGURE\_TYPE

def \_\_repr\_\_(self):

return '{}, цвет - {}, сторона = {}, площадь = {}.'.format(

self.get\_type(),

self.color.colorproperty,

self.side,

self.get\_square()

)

*rectangle.py*

from lab\_python\_oop.geometrical\_figure import GeometricalFigure

from lab\_python\_oop.f\_color import FigurColor

class Rectangle(GeometricalFigure):

FIGURE\_TYPE = "Прямоугольник"

@classmethod

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

self.width = width

self.height = height

self.color = FigurColor()

self.color.colorproperty = color

def get\_type(self):

return self.FIGURE\_TYPE

def get\_square(self):

return self.width\*self.height

def \_\_repr\_\_(self):

return '{}, цвет - {}, ширина = {}, высота = {}, площадь = {}.'.format(

self.get\_type(),

self.color.colorproperty,

self.width,

self.height,

self.get\_square()

)

Тесты

