**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 ИУ5-51б  Афанасьев Даниил Миронович |  | 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.box import \*  
from lab\_python\_oop.rectangle import \*  
from lab\_python\_oop.circle import \*  
import numpy as np  
  
def main():  
 r = Rectangle("синего", 2, 2)  
 c = Circle("зеленого", 2)  
 s = Box("красного", 2)  
 print(r)  
 print(c)  
 print(s)  
  
 mas = np.array([[1,5,6,5],  
 [3,9,-2,0],  
 [9,4,7,1],  
 [1,0,9,4]])  
  
 print(f"\n{np.linalg.det(mas)}")  
 print(mas.dot(mas))  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

box.py

from lab\_python\_oop.rectangle import Rectangle  
  
class Box(Rectangle):  
 *"""  
 Класс «Квадрат» наследуется от класса «Прямоугольник».  
 """* FIGURE\_TYPE = "Квадрат"  
 def \_\_init\_\_(self, color, width):  
 super().\_\_init\_\_(color, width, width)  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGURE\_TYPE  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета длиной {} площадью {}.'.format(  
 Box.get\_figure\_type(),  
 self.color,  
 self.width,  
 self.square()  
 )

circle.py

from lab\_python\_oop.geometric\_shape import Geometric\_shape  
from lab\_python\_oop.color import Color  
from math import pi  
  
class Circle(Geometric\_shape):  
 *"""  
 Класс «Круг» наследуется от класса «Геометрическая фигура».  
 """* FIGURE\_TYPE = "Круг"  
 def \_\_init\_\_(self, color, radius):  
 self.radius = radius  
 self.color = Color()  
 self.color = color  
  
 def square(self):  
 return pi \* self.radius \*\* 2  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGURE\_TYPE  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета радиуса {} и площадью {}.'.format(  
 Circle.get\_figure\_type(),  
 self.color,  
 self.radius,  
 self.square()  
 )

color.py

class Color:  
 *"""  
 Класс «Цвет фигуры»  
 """* def \_\_init\_\_(self):  
 self.\_color = None  
  
 @property  
 def color(self):  
 return self.\_color  
  
 @color.setter  
 def color(self, value):  
 self.\_color = value  
  
 @color.deleter  
 def color(self):  
 del self.\_color

geometric\_shape.py

from abc import ABC, abstractmethod  
  
  
class Geometric\_shape(ABC):  
 *"""  
 Класс «Геометрическая фигура».  
 """* FIGURE\_TYPE = "Геометрическая фигура"  
  
 @abstractmethod  
 def square(self):  
 pass

rectangle.py

from lab\_python\_oop.geometric\_shape import Geometric\_shape  
from lab\_python\_oop.color import Color  
  
class Rectangle(Geometric\_shape):  
 *"""  
 Класс «Прямоугольник» наследуется от класса «Геометрическая фигура».  
 """* FIGURE\_TYPE = "Прямоугольник"  
 def \_\_init\_\_(self, color, width, height):  
 self.width = width  
 self.height = height  
 self.color = Color()  
 self.color = color  
  
 def square(self):  
 return self.width \* self.height  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGURE\_TYPE  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(  
 Rectangle.get\_figure\_type(),  
 self.color,  
 self.width,  
 self.height,  
 self.square()  
 )

**Анализ результатов**

