**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет “Радиотехнический”

Кафедра “Системы обработки информации и управления”

Курс «Парадигмы и конструкции языка»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

Вариант №14

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б: |  | преподаватель каф. ИУ5 |
| Сахарова О.П |  | Гапанюк Ю.Е. |
|  |  |  |

Москва, 2025 г.

**Описание задания**

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.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь.
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 - https://pyformat.info/
* 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 (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**

**main.py**  
from colorama import Fore, init  
from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.square import Square  
  
  
def main():  
 init(autoreset=True)  
 N = 14  
  
 rectangle = Rectangle(N, N, "синего")  
 circle = Circle(N, "зеленого")  
 square = Square(N, "красного")  
  
 print(Fore.BLUE + str(rectangle))  
 print(Fore.GREEN + str(circle))  
 print(Fore.RED + str(square))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**abs\_figure.py**

from abc import ABC, abstractmethod  
  
@property  
@abstractmethod  
def name(self):  
 pass  
  
class Figure(ABC):  
 @abstractmethod  
 def S(self):  
 pass  
  
 @abstractmethod  
 def \_\_repr\_\_(self):  
 pass

**color.py**

class FigureColor:  
 def \_\_init\_\_(self):  
 self.\_color = None  
  
 @property  
 def color(self):  
 return self.\_color  
  
 @color.setter  
 def color(self, value):  
 self.\_color = value  
  
 def get\_color(self):  
 return self.\_color  
  
 def set\_color(self, value):  
 self.\_color = value

**rectangle.py**

from .abs\_figure import Figure  
from .color import FigureColor  
  
  
class Rectangle(Figure):  
 def \_\_init\_\_(self, width, height, color):  
 self.\_width = width  
 self.\_height = height  
 self.\_color\_property = FigureColor()  
 self.\_color\_property.color = color  
 self.\_name = "Прямоугольник"  
  
 @property  
 def name(self):  
 return self.\_name  
  
 def S(self):  
 return self.\_width \* self.\_height  
  
 def \_\_repr\_\_(self):  
 return "{} {} цвета шириной {} и высотой {}, площадью {}.".format(  
 self.\_name,  
 self.\_color\_property.color,  
 self.\_width,  
 self.\_height,  
 self.S()  
 )

**circle.py**

from math import pi  
from .abs\_figure import Figure  
from .color import FigureColor  
  
  
class Circle(Figure):  
 def \_\_init\_\_(self, radius, color):  
 self.\_radius = radius  
 self.\_color\_property = FigureColor()  
 self.\_color\_property.color = color  
 self.\_name = "Круг"  
  
 @property  
 def name(self):  
 return self.\_name  
  
 def S(self):  
 return pi \* self.\_radius \*\* 2  
  
 def \_\_repr\_\_(self):  
 return "{} {} цвета радиусом {}, площадью {:.2f}.".format(  
 self.\_name,  
 self.\_color\_property.color,  
 self.\_radius,  
 self.S()  
 )

**square.py**

from .rectangle import Rectangle  
  
  
class Square(Rectangle):  
 def \_\_init\_\_(self, side, color):  
 super().\_\_init\_\_(side, side, color)  
 self.\_name = "Квадрат"  
  
 @property  
 def name(self):  
 return self.\_name  
  
 def \_\_repr\_\_(self):  
 return "{} {} цвета со стороной {}, площадью {}.".format(  
 self.\_name,  
 self.\_color\_property.color,  
 self.\_width,  
 self.S()  
 )

**Примеры выполнения программы**

**Изображение выглядит как текст, программное обеспечение, Шрифт, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.**