**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| Студент группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Абрамов А. В. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 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.
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.](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»:
   * 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 (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.
11. **Дополнительное задание.** Протестируйте корректность работы Вашей программы с помощью модульного теста.

**Текст программы**

**main.py**

import numpy as np  
  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.square import Square  
  
  
def main():  
 rectangle = Rectangle(5, 10, "Biege")  
 print(rectangle.repr())  
 circle = Circle(10, "Black")  
 print(circle)  
 square = Square(4, "Cyan")  
 print(square)  
 matrix = np.array(([10, 15, 9],  
 [5, 7, 1]))  
 print(matrix)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**lab\_python\_oop/geom\_figure.py**

from abc import ABC, abstractmethod  
  
  
class Figure(ABC):  
 @abstractmethod  
 def area(self):  
 pass

**lab\_python\_oop/color.py**

from abc import ABC  
  
  
class Color(ABC):  
 def \_\_init\_\_(self, color\_):  
 self.\_\_color = color\_  
  
 def get\_color(self):  
 return self.\_\_color  
  
 def set\_color(self, color\_):  
 self.\_\_color = color\_  
  
 color = property(get\_color, set\_color)

**lab\_python\_oop/rectangle.py**

from lab\_python\_oop.color import Color  
from lab\_python\_oop.geom\_figure import Figure  
  
  
class Rectangle(Figure):  
 def \_\_init\_\_(self, length\_, width\_, color\_):  
 self.length = length\_  
 self.width = width\_  
 self.my\_color = Color(color\_)  
 self.name = "Прямоугольник"  
  
 def area(self):  
 return self.width \* self.length  
  
 def get\_name(self):  
 return self.name  
  
 def repr(self):  
 return "Фигура: {}, Длина - {}, Ширина - {}, Цвет - {}, Площадь - {}".format(self.get\_name(),  
 self.length,  
 self.width,  
 self.my\_color.color,  
 self.area())

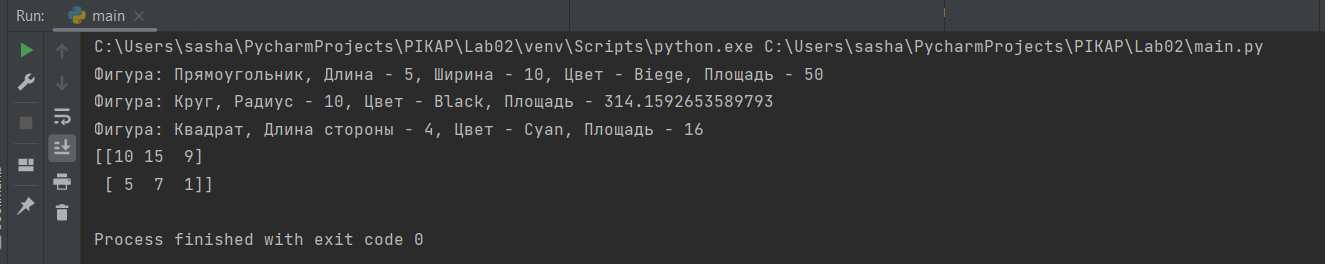
**lab\_python\_oop/circle.py**

from lab\_python\_oop.color import Color  
from lab\_python\_oop.geom\_figure import Figure  
import math  
  
  
class Circle(Figure):  
 def \_\_init\_\_(self, radius\_, color\_):  
 self.radius = radius\_  
 self.my\_color = Color(color\_)  
 self.name = "Круг"  
  
 def area(self):  
 return math.pi \* self.radius \* self.radius  
  
 def get\_name(self):  
 return self.name  
  
 def repr(self):  
 return "Фигура: {}, Радиус - {}, Цвет - {}, Площадь - {}".format(self.get\_name(), self.radius,  
 self.my\_color.color, self.area())

**lab\_python\_oop/square.py**

from lab\_python\_oop.color import Color  
from lab\_python\_oop.rectangle import Rectangle  
  
  
class Square(Rectangle):  
 def \_\_init\_\_(self, side\_, color\_):  
 self.side = side\_  
 self.my\_color = Color(color\_)  
 self.name = "Квадрат"  
  
 def area(self):  
 return self.side \* self.side  
  
 def get\_name(self):  
 return self.name  
  
 def \_\_repr\_\_(self):  
 return "Фигура: {}, Длина стороны - {}, Цвет - {}, Площадь - {}".format(self.get\_name(), self.side,  
 self.my\_color.color, self.area())

**Пример выполнения программы**

****