**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «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 РТ5-31Б: |  | преподаватель кафедры ИУ5 |
| Фёдоров И. А. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

Постановка задачи

**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 [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»:
   * 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.
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.

Текст программы

Файл «Shape.py»

from abc import ABCMeta, abstractmethod  
  
  
class Shape:  
 \_\_metaclass\_\_ = ABCMeta  
  
 @abstractmethod  
 def square(self):  
 *"""площадь"""* @abstractmethod  
 def \_\_repr\_\_(self):  
 *"""параметры"""*

Файл «Rectangle.py»

from Color import ShapeColor  
from Shape import Shape  
  
  
class Rectangle(Shape):  
 def \_\_init\_\_(self, a, b, color):  
 self.a = a  
 self.b = b  
 self.color = ShapeColor(color.r, color.g, color.b)  
 self.title = 'Rectangle'  
  
 def square(self):  
 return self.a \* self.b  
  
 def \_\_repr\_\_(self):  
 return f"""title: {self.title}; (a,b): ({self.a}, {self.b}); color(r, g, b):  
 ({self.color.r}, {self.color.g}, {self.color.b}); square:{self.square()}"""

Файл «Color.py»

class ShapeColor:  
 def \_\_init\_\_(self, r, g, b):  
 self.r = r  
 self.g = g  
 self.b = b

Файл «Square.py»

from Rectangle import Rectangle  
  
  
class Square(Rectangle):  
 def \_\_init\_\_(self, a, color):  
 super().\_\_init\_\_(a, a, color)  
 self.title = 'Square'  
  
 def \_\_repr\_\_(self):  
 return f"""title: {self.title}; a = {self.a}; color(r, g, b):  
 ({self.color.r}, {self.color.g}, {self.color.b}); square:{self.square()}"""

Файл «Circle.py»

import math  
  
from Color import ShapeColor  
from Shape import Shape  
  
  
class Circle(Shape):  
 def \_\_init\_\_(self, r, color):  
 self.r = r  
 self.color = ShapeColor(color.r, color.g, color.b)  
 self.title = 'Circle'  
  
 def square(self):  
 return math.pi \* self.r \*\* 2  
  
 def \_\_repr\_\_(self):  
 return f"""title: {self.title}; r = {self.r}; color(r, g, b):  
 ({self.color.r}, {self.color.g}, {self.color.b}); square:{self.square()}"""

Файл «main.py»

from Circle import Circle  
from Color import ShapeColor  
from Rectangle import Rectangle  
from Square import Square  
  
  
def main():  
 color = ShapeColor(1, 2, 3)  
 rectangle = Rectangle(1, 2, color)  
 circle = Circle(1, color)  
 square = Square(1, color)  
  
 print(rectangle.\_\_repr\_\_())  
 print(circle.\_\_repr\_\_())  
 print(square.\_\_repr\_\_())  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

Экранные формы с примерами выполнения программы

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание