**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил:  Макеева Е. А.  ИУ5-31Б |  | Проверил:  Нардид А. Н. |
| Подпись и дата: |  | Подпись и дата: |

Постановка задачи

**Цель лабораторной работы:** изучение объектно-ориентированных возможностей языка Python.

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.

Код программы

circle.py

import math  
from geometric\_figure import GeometricFigure  
from figure\_color import FigureColor  
  
  
class Circle(GeometricFigure):  
 name = "Круг"  
  
 def \_\_init\_\_(self, radius, color):  
 self.radius = radius  
 self.color = FigureColor(color)  
  
 def area(self):  
 return math.pi \* self.radius \*\* 2  
  
 def \_\_repr\_\_(self):  
 return "Фигура: {0}, Цвет: {1}, Радиус: {2}, Площадь: {3:.2f}".format(  
 self.name, self.color.color(), self.radius, self.area()  
 )

figure\_color.py

class FigureColor:  
 def \_\_init\_\_(self, color):  
 self.\_color = color  
  
 def color(self):  
 return self.\_color

geometric\_figure.py

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

rectangle.py

from geometric\_figure import GeometricFigure  
from figure\_color import FigureColor  
  
  
class Rectangle(GeometricFigure):  
 name = "Прямоугольник"  
  
 def \_\_init\_\_(self, width, height, color):  
 self.width = width  
 self.height = height  
 self.color = FigureColor(color)  
  
 def area(self):  
 return self.width \* self.height  
  
 def \_\_repr\_\_(self):  
 return "Фигура: {0}, Цвет: {1}, Ширина: {2}, Высота: {3}, Площадь: {4}".format(  
 self.name, self.color.color(), self.width, self.height, self.area()  
 )

square.py

from rectangle import Rectangle  
  
  
class Square(Rectangle):  
 name = "Квадрат"  
  
 def \_\_init\_\_(self, side, color):  
 super().\_\_init\_\_(side, side, color)  
  
 def \_\_repr\_\_(self):  
 return "Фигура: {0}, Цвет: {1}, Сторона: {2}, Площадь: {3}".format(  
 self.name, self.color.color(), self.width, self.area()  
 )

main.py

from rectangle import Rectangle  
from circle import Circle  
from square import Square  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 N = 13  
  
 rectangle = Rectangle(N, N, "синий")  
 print(rectangle)  
  
 circle = Circle(N, "зеленый")  
 print(circle)  
  
 square = Square(N, "красный")  
 print(square)  
  
 import datetime  
  
 print("Текущее время:", datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S'))

**Анализ результатов**

