**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 ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Канаева Д.Ч. |  | Нардид А.Н. |
| Подпись и дата: |  | Подпись и дата: |

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

**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.
11. **Дополнительное задание.** Протестируйте корректность работы Вашей программы с помощью модульного теста.

**Текст программы:**

**lab\_python\_oop\GF.py**

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

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

class Color:  
 def \_\_init\_\_(self, color\_):  
 self.color = color\_  
  
 def \_\_repr\_\_(self):  
 return self.color

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

from lab\_python\_oop.GF import GeometricFigure  
from lab\_python\_oop.color import Color  
import math  
  
class Circle(GeometricFigure):  
 def \_\_init\_\_(self, rad\_, color):  
 self.radius = rad\_  
 self.color = Color(color)  
 self.type = "Круг"  
  
 def area(self):  
 return math.pi \* self.radius \*\* 2  
  
 def \_\_repr\_\_(self):  
 return "Circle(type={}, radius={}, color={}, area={})".format(self.type, self.radius, self.color, self.area())

**lab\_python\_oop\rect.py**

from lab\_python\_oop.color import Color  
from lab\_python\_oop.GF import GeometricFigure  
  
class Rectangle(GeometricFigure):  
 def \_\_init\_\_(self, width\_, height\_, color):  
 self.width = width\_  
 self.height = height\_  
 self.color = Color(color)  
 self.type = "Прямоугольник"  
  
 def area(self):  
 return self.width \* self.height  
  
 def \_\_repr\_\_(self):  
 return "Rectangle(type={}, width={}, height={}, color={}, area={})".format(self.type, self.width, self.height, self.color, self.area())

**lab\_python\_oop\sq.py**

from lab\_python\_oop.rect import Rectangle  
from lab\_python\_oop.color import Color  
  
class Square(Rectangle):  
 def \_\_init\_\_(self, length\_, color):  
 self.length = length\_  
 self.color = Color(color)  
 self.type = "Квадрат"  
  
 def area(self):  
 return self.length \*\* 2  
  
 def \_\_repr\_\_(self):  
 return "Square(type={}, length={}, color={}, area={})".format(self.type, self.length, self.color, self.area())

**main.py**

from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.rect import Rectangle  
from lab\_python\_oop.sq import Square  
import matplotlib.pyplot as plt  
  
rectangle = Rectangle(8, 8, "синий")  
circle = Circle(8, "зеленый")  
square = Square(8, "красный")  
  
print(rectangle, '\n', circle, '\n', square)  
  
#данные  
x = [1, 2, 3, 4, 5]  
y = [2, 4, 6, 8, 10]  
  
#построение графика  
plt.plot(x, y)  
  
#настройка осей  
plt.xlabel('x')  
plt.ylabel('y')  
plt.title('пример графика')  
  
#отображение графика в файле  
plt.savefig('lab\_python\_oop\picture.png')

**Экранные формы с примерами выполнения программы:**



