**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-52Б |  | преподаватель каф. ИУ5 |
| Корчевский Александр |  | Гапанюк Ю.Е. |
|  |  |  |

Москва, 2021 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.](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:**

**main.py:**

from lab\_python\_oop.circle import Circle

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.square import Square

def main():

rec = Rectangle('красный', 2, 3)

circ = Circle('розовый', 4)

sq = Square('синий', 5)

print(rec)

print(circ)

print(sq)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Figure.py:**

from abc import ABC, abstractmethod

class Figure(ABC):

@abstractmethod

def calcSquare(self):

pass

**Color.py:**

class Color:

def \_\_init\_\_(self):

self.\_color = None

@property

def color(self):

return self.\_color

@color.setter

def color(self, value):

self.\_color = value

**Rectangle.py:**

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.color import Color

class Rectangle(Figure):

figure\_type = 'Прямоугольник'

@classmethod

def get\_figure\_type(cls):

return cls.figure\_type

def \_\_init\_\_(self, color\_par, width\_par, height\_par):

self.width = width\_par

self.height = height\_par

self.fcolor = Color()

self.fcolor.color = color\_par

def calcSquare(self):

return self.height \*self.width

def \_\_repr\_\_(self) -> str:

return '{} {} высотой {} шириной {} и площадью {}'.format(Rectangle.get\_figure\_type(), self.fcolor.color, self.height, self.width, self.calcSquare())

**Circle.py:**

from lab\_python\_oop.figure import Figure

from lab\_python\_oop.color import Color

from math import pi

class Circle(Figure):

figure\_type = 'Круг'

@classmethod

def get\_figure\_type(cls) -> str:

return cls.figure\_type

def \_\_init\_\_(self, color\_par, radius\_par):

self.radius = radius\_par

self.fcolor = Color()

self.fcolor.color = color\_par

def calcSquare(self):

return self.radius \*self.radius \*pi

def \_\_repr\_\_(self) -> str:

return '{} {} радиуса {} и площадью {}'.format(Circle.get\_figure\_type() ,self.fcolor.color, self.radius, round(self.calcSquare(), 3))

**Square.py:**

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

figure\_type = 'Квадрат'

@classmethod

def get\_figure\_type(cls):

return cls.figure\_type

def \_\_init\_\_(self, color\_par, width\_par):

Rectangle.\_\_init\_\_(self, color\_par, width\_par, width\_par)

def \_\_repr\_\_(self) -> str:

return '{} {} высотой {} шириной {} и площадью {}'.format(Square.get\_figure\_type(), self.fcolor.color, self.height, self.width, self.calcSquare())

**Экранные формы с примерами выполнения программы:**

|  |  |
| --- | --- |
| *Пример* | *Вывод программы* |
| 1 |  |