Московский государственный технический

университет им. Н.Э. Баумана.

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python.»

Выполнил:   
студент группы ИУ5-35Б   
Акулова А.А

Подпись и дата:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

28.12.2021

Проверил:  
преподаватель кафедры ИУ5  
Гапанюк Ю. Е.

Подпись и дата:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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.
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.
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:**

import pygame  
pygame.init()  
from lab\_python\_oop.circle import circle  
from lab\_python\_oop.ShapeColor import ShapeColor  
from lab\_python\_oop.rectangle import rectangle  
from lab\_python\_oop.square import square  
from lab\_python\_oop.geometry\_shape import geometry\_shape  
  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 blue = ShapeColor(0, 0, 255)  
 green = ShapeColor(0, 255, 0)  
 red = ShapeColor(255, 0, 0)  
  
 Rectangle = rectangle(1, 1, blue)  
 Circle = circle(1, red)  
 Square = square(1, green)  
  
  
 Rectangle.Repr()  
 Circle.Repr()  
 Square.Repr()

**Текст программы circle:**

from lab\_python\_oop.geometry\_shape import geometry\_shape  
from lab\_python\_oop.ShapeColor import ShapeColor  
from math import pi  
import pygame  
  
  
class circle(geometry\_shape):  
 r = float()  
 color = ShapeColor(0, 0, 0)  
 name = "Circle"  
  
 def \_\_init\_\_(self, r, color):  
 self.r = r  
 self.color = color  
  
 def Repr(self):  
 resultStr = self.name + ": \n"  
 resultStr += " radius = {}".format(self.r)  
 resultStr += " color = {}.{}.{}".format(self.color.r, self.color.g, self.color.b)  
 print(resultStr)  
  
 def Square(self):  
 return pi \* self.r \*\* 2

**Текст программы rectangle:**

from lab\_python\_oop.geometry\_shape import geometry\_shape  
from lab\_python\_oop.ShapeColor import ShapeColor  
  
  
class rectangle(geometry\_shape):  
 width = float()  
 height = float()  
 color = ShapeColor(0, 0, 0)  
 name = "Rectangle"  
  
 def Repr(self):  
 resultStr = self.name + ": \n"  
 resultStr += " width = {}".format(self.width)  
 resultStr += " height = {}".format(self.height)  
 resultStr += " color = {}.{}.{}".format(self.color.r, self.color.g, self.color.b)  
 print(resultStr)  
  
 def Square(self):  
 return self.width \* self.height  
  
 def \_\_init\_\_(self, width, height, color):  
 self.color = color  
 self.width = width  
 self.height = height

**Текст программы ShapeColor:**

class ShapeColor():  
 r = int(0)  
 g = int(0)  
 b = int(0)  
  
 def \_\_init\_\_(self, r, g, b):  
 self.r = r  
 self.g = g  
 self.b = b

**Текст программы square:**

from lab\_python\_oop.rectangle import rectangle  
  
class square(rectangle):  
 name = "Square"  
  
 def Repr(self):  
 resultStr = self.name + ": \n"  
 resultStr += " side = {}".format(self.width)  
 resultStr += " color = {}.{}.{}".format(self.color.r, self.color.g, self.color.b)  
 print(resultStr)  
  
 def \_\_init\_\_(self, side, color):  
 self.width = side  
 self.height = side  
 self.color = color

**Текст программы geometry\_shape:**

from abc import ABC, abstractmethod  
  
class geometry\_shape(ABC):  
 @abstractmethod  
 def Square(self):  
 pass

**Экранная форма с примерами выполнения программы:**

