**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе № 2

« Объектно-ориентированные возможности языка Python »

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Зелинский Даниил Михайлович |  | Гапанюк Юрий Евгеньевич |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

## Задание.

Разработать программу для решения биквадратного уравнения.

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»:
   * Определить метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Использовать метод format.
   * Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
10. В корневом каталоге проекта создать файл main.py для тестирования Ваших классов. Создайте следующие объекты и выведите о них информацию в консоль (N=6):
    * 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.

import numpy as np  
from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.square import Square  
  
  
def main():  
 rect=Rectangle("синего", 6, 6)  
 circ=Circle('зелёного', 6)  
 squa=Square('красного', 6)  
  
 ar = np.array([rect, circ, squa])  
 print (ar[0:3])  
  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

from abc import ABC, abstractmethod  
  
class Figure(ABC):  
 #абстрактный класс геометрическая фигура  
 @abstractmethod  
 def square(selfself):  
 #виртуальный метод вычисления площади  
 pass

class FigureColor:  
 #класс цвет фигуры  
 def \_\_init\_\_(self):  
 self.\_color=None  
  
 @property  
 def colorproperty(self):  
 #геттер  
 return self.\_color  
  
 @colorproperty.setter  
 def colorproperty(self,value):  
 #сеттер  
 self.\_color=value

from lab\_python\_oop.figure import Figure  
from lab\_python\_oop.color import FigureColor  
  
class Rectangle(Figure):  
 #прямоугольник - наследник фигуры  
  
 FIGURE\_TYPE="Прямоугольник"  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGURE\_TYPE  
  
 def \_\_init\_\_(self, color\_param, width\_param, height\_param):  
 #объект класса FigureColor создаётся здесь для хранения цвета  
 self.width=width\_param  
 self.height=height\_param  
 self.fc=FigureColor()  
 self.fc.colorproperty=color\_param  
  
 def square(self):  
 #переопределение метода вычисления площади  
 return self.width\*self.height  
  
 def \_\_repr\_\_(self):  
 #объект класса в виде строки  
 return '{} {} цвета шириной {}, высотой {} и площадью {}'.format(  
 Rectangle.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.width,  
 self.height,  
 self.square()  
 )

from lab\_python\_oop.rectangle import Rectangle  
  
class Square (Rectangle):  
 FIGUTE\_TYPE= "Квадрат"  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGUTE\_TYPE  
  
 def \_\_init\_\_(self, color\_param, side\_param):  
 self.side=side\_param  
 Rectangle.\_\_init\_\_(self,color\_param, self.side, self.side)  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета со стороной {} и площадью {}'.format(  
 Square.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.side,  
 self.square()  
 )

from lab\_python\_oop.figure import Figure  
from lab\_python\_oop.color import FigureColor  
import math  
  
class Circle (Figure):  
  
 FIGURE\_TYPE="Круг"  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGURE\_TYPE  
  
 def \_\_init\_\_(self, color\_param, r\_param):  
 self.r=r\_param  
 self.fc=FigureColor()  
 self.fc.colorproperty=color\_param  
  
 def square(self):  
 return math.pi\*(self.r\*\*2)  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета радиусом {} и площадью {}'.format(  
 Circle.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.r,  
 self.square()  
 )

## Примеры.

