**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: |  | Проверил: |
| студент группы ИУ5-33 |  | преподаватель каф. ИУ5 |
| Семенов Вячеслав |  | Канев Антон |
|  |  |

2021 г.

1. **Описание задания**
2. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.
3. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
4. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
5. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
6. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры.
7. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
8. 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.
9. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math
10. 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
    * Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
11. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов. 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.

**2. Текст программы**

main.py

from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.square import Square  
from PIL import Image, ImageDraw  
  
def figuresDraw(radius, side, height):  
 image = Image.new("RGBA", (620, 620), (0, 0, 0, 0))  
 draw = ImageDraw.Draw(image)  
 draw.ellipse((0, 0, radius \* 100, radius \* 100), fill='green', outline='green')  
 draw.rectangle((0, 0, side \* 100, side \* 100), fill='red', outline='red')  
 draw.rectangle((0, 0, side \* 100, height \* 100), fill='blue', outline='blue')  
 image.show()  
  
def main():  
 side = 5  
 height = 3  
 radius = 15  
 r = Rectangle("синего", side, height)  
 c = Circle("зеленого", radius)  
 s = Square("красного", side)  
 print(r)  
 print(c)  
 print(s)  
 figuresDraw(radius, side, height)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

figure.py

from abc import ABC, abstractmethod  
  
  
class Figure(ABC):  
 *"""  
 Абстрактный класс «Геометрическая фигура»  
 """* @abstractmethod  
 def square(self):  
 *"""  
 содержит виртуальный метод для вычисления площади фигуры.  
 """* pass

color.py

class FigureColor:  
 *"""  
 Класс «Цвет фигуры»  
 """* def \_\_init\_\_(self):  
 self.\_color = None  
  
 @property  
 def colorproperty(self):  
 *"""  
 Get-аксессор  
 """* return self.\_color  
  
 @colorproperty.setter  
 def colorproperty(self, value):  
 *"""  
 Set-аксессор  
 """* self.\_color = value

rectangle.py

from lab\_python\_oop.figure import Figure  
from lab\_python\_oop.color import FigureColor  
  
  
class Rectangle(Figure):  
 *"""  
 Класс «Прямоугольник» наследуется от класса «Геометрическая фигура».  
 """* FIGURE\_TYPE = "Прямоугольник"  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGURE\_TYPE  
  
 def \_\_init\_\_(self, color\_param, width\_param, height\_param):  
 *"""  
 Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет».  
 В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.  
 """* self.width = width\_param  
 self.height = height\_param  
 self.fc = FigureColor()  
 self.fc.colorproperty = color\_param  
  
 def square(self):  
 *"""  
 Класс должен переопределять метод, вычисляющий площадь фигуры.  
 """* return self.width\*self.height  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(  
 Rectangle.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.width,  
 self.height,  
 self.square()  
 )

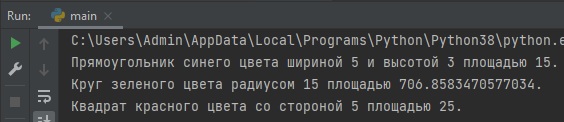
circle.py

from lab\_python\_oop.figure import Figure  
from lab\_python\_oop.color import FigureColor  
import math  
  
  
class Circle(Figure):  
 *"""  
 Класс «Круг» наследуется от класса «Геометрическая фигура».  
 """* FIGURE\_TYPE = "Круг"  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGURE\_TYPE  
  
 def \_\_init\_\_(self, color\_param, r\_param):  
 *"""  
 Класс должен содержать конструктор по параметрам «радиус» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.  
 """* self.r = r\_param  
 self.fc = FigureColor()  
 self.fc.colorproperty = color\_param  
  
 def square(self):  
 *"""  
 Класс должен переопределять метод, вычисляющий площадь фигуры.  
 """* return math.pi\*(self.r\*\*2)  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета радиусом {} площадью {}.'.format(  
 Circle.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.r,  
 self.square()  
 )

square.py

from lab\_python\_oop.rectangle import Rectangle  
  
  
class Square(Rectangle):  
 *"""  
 Класс «Квадрат» наследуется от класса «Прямоугольник».  
 """* FIGURE\_TYPE = "Квадрат"  
  
 @classmethod  
 def get\_figure\_type(cls):  
 return cls.FIGURE\_TYPE  
  
 def \_\_init\_\_(self, color\_param, side\_param):  
 *"""  
 Класс должен содержать конструктор по параметрам «сторона» и «цвет».  
 """* self.side = side\_param  
 super().\_\_init\_\_(color\_param, self.side, self.side)  
  
 def \_\_repr\_\_(self):  
 return '{} {} цвета со стороной {} площадью {}.'.format(  
 Square.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.side,  
 self.square()  
 )

**3. Экранные формы с примерами выполнения программ**

****

