**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Казовые компоненты интернет технологий»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Руководитель: |
| студент группы ИУ5-33Б: |  | преподаватель каф. ИУ5 |
| Журмилов В. Д. |  | Гапанюк Ю.Е. |
|  |  |  |

Москва, 2022 г.

**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.

6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь.

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»:

10. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - https://pyformat.info/

11. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.

12. 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.

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

**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  
  
  
def main():  
 r = Rectangle("синего", 3, 2)  
 c = Circle("зеленого", 5)  
 s = Square("красного", 5)  
 print(r)  
 print(c)  
 print(s)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**test.py**

import unittest  
  
from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.square import Square  
import math  
  
  
class TestRectangle(unittest.TestCase):  
 def test\_rect\_init(self):  
 r = Rectangle("синего", 12, 12)  
 self.assertEqual(r.get\_figure\_type(), 'Прямоугольник')  
 self.assertEqual(r.fc.colorproperty, 'синего')  
 self.assertEqual(r.width, 12)  
 self.assertEqual(r.height, 12)  
 self.assertEqual(r.square(), r.height \* r.width)  
  
 def test\_rect\_repr(self):  
 r = Rectangle("синего", 12, 12)  
 self.assertEqual(r.\_\_repr\_\_(), 'Прямоугольник синего цвета шириной 12 и высотой 12 площадью 144.')  
  
  
class TestCircle(unittest.TestCase):  
 def test\_circ\_init(self):  
 c = Circle("зеленого", 12)  
 self.assertEqual(c.get\_figure\_type(), 'Круг')  
 self.assertEqual(c.fc.colorproperty, 'зеленого')  
 self.assertEqual(c.r, 12)  
 self.assertEqual(c.square(), math.pi \* (c.r \*\* 2))  
  
 def test\_circ\_repr(self):  
 c = Circle("зеленого", 12)  
 self.assertEqual(c.\_\_repr\_\_(), 'Круг зеленого цвета радиусом 12 площадью 452.3893421169302.')  
  
  
class TestSquare(unittest.TestCase):  
 def test\_sq\_init(self):  
 s = Square("красного", 12)  
 self.assertEqual(s.get\_figure\_type(), 'Квадрат')  
 self.assertEqual(s.fc.colorproperty, 'красного')  
 self.assertEqual(s.width, 12)  
 self.assertEqual(s.height, 12)  
 self.assertEqual(s.square(), s.height \* s.width)  
  
 def test\_sq\_repr(self):  
 s = Square("красного", 12)  
 self.assertEqual(s.\_\_repr\_\_(), 'Квадрат красного цвета со стороной 12 площадью 144.')  
  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

**Примеры выполнения:**

